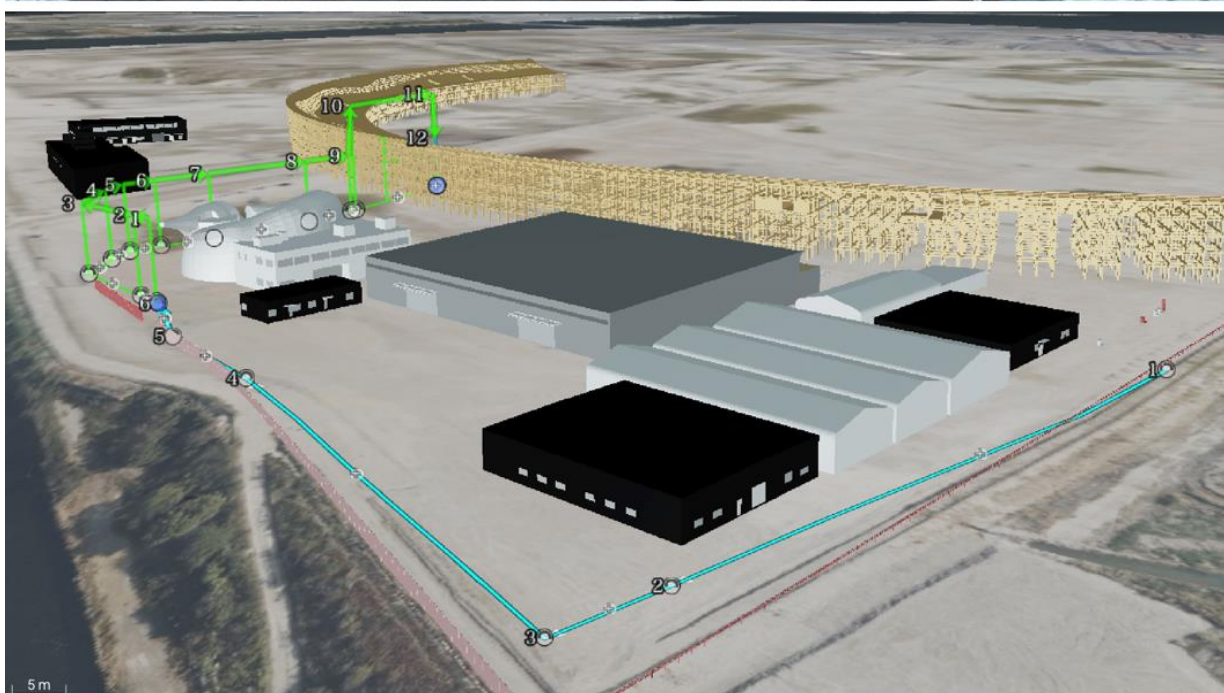




3D都市モデルとBIMを活用した モビリティ自律運航システム v3.0 技術検証レポート

Technical Report on Autonomous Mobility Operation System Utilizing 3D City Models and BIM Models v3.0

series
No.

97

目次

1. ユースケースの概要	- 4 -
1-1. 現状と課題	- 4 -
1-2. 課題解決のアプローチ	- 5 -
1-3. 創出価値	- 5 -
1-4. 想定事業機会	- 6 -
2. 実証実験の概要	- 7 -
2-1. 実証仮説	- 7 -
2-2. 実証フロー	- 7 -
2-3. 検証ポイント	- 7 -
2-4. 実施体制	- 8 -
2-5. 実証エリア	- 9 -
2-6. スケジュール	- 10 -
3. 開発スコープ	- 11 -
3-1. 概要	- 11 -
3-2. 開発内容	- 11 -
4. 実証システム	- 13 -
4-1. アーキテクチャ	- 13 -
4-1-1. システムアーキテクチャ	- 13 -
4-1-2. データアーキテクチャ	- 16 -
4-1-3. ハードウェアアーキテクチャ	- 19 -
4-2. システム機能	- 32 -
4-2-1. システム機能一覧	- 32 -
4-2-2. 利用したソフトウェア・ライブラリ	- 34 -
4-2-3. 開発機能の詳細要件	- 37 -
4-3. アルゴリズム	- 82 -
4-3-1. 利用したアルゴリズム	- 82 -
4-3-2. 開発したアルゴリズム	- 86 -
4-4. データインタフェース	- 90 -
4-4-1. ファイル入力インタフェース	- 90 -
4-4-2. ファイル出力インタフェース	- 90 -
4-4-3. 内部連携インタフェース	- 92 -
4-4-4. 外部連携インタフェース	- 99 -
4-5. 実証に用いたデータ	- 102 -
4-5-1. 活用したデータ一覧	- 102 -
4-5-2. 生成・変換したデータ	- 104 -
4-6. ユーザーインタフェース	- 106 -

4-6-1. 画面一覧	- 106 -
4-6-2. 画面遷移図	- 108 -
4-6-3. 各画面仕様詳細	- 109 -
4-7. 実証システムの利用手順	- 128 -
4-7-1. 実証システムの利用フロー	- 128 -
4-7-2. 各画面操作方法	- 130 -
5. システムの非機能要件	- 140 -
5-1. 社会実装に向けた非機能要件	- 140 -
5-2. 実証観点での非機能要件	- 143 -
6. 品質	- 146 -
6-1. 機能要件の品質担保	- 146 -
6-2. 非機能要件の品質担保	- 147 -
7. 実証技術の機能要件の検証	- 149 -
7-1. 水平・垂直方向の自動運航の検証	- 167 -
7-1-1. 検証目的	- 167 -
7-1-2. KPI	- 167 -
7-1-3. 検証方法と検証シナリオ	- 168 -
7-1-4. 検証結果	- 169 -
7-2. RTK 測位データを活用した車両の自己位置精度向上の検証	- 149 -
7-2-1. 検証目的	- 149 -
7-2-2. KPI	エラー! ブックマークが定義されていません。
7-2-3. 検証方法と検証シナリオ	- 150 -
7-2-4. 検証結果	- 155 -
7-3. SENSYN CORE Pilot を用いた車両のルート作成（調査研究項目）	- 167 -
7-3-1. 検証目的	エラー! ブックマークが定義されていません。
7-3-2. 検証方法	エラー! ブックマークが定義されていません。
7-3-3. 検証結果	エラー! ブックマークが定義されていません。
8. 実証技術の非機能要件の検証	- 176 -
8-1. 検証目的	- 176 -
8-2. KPI	- 176 -
8-2-1. 検証方法と検証シナリオ	- 177 -
8-2-2. 検証結果	- 177 -
9. BtoB ビジネスでの有用性検証	- 178 -
9-1. 検証目的	- 178 -
9-2. 検証方法	- 179 -
9-3. 被験者	- 179 -
9-4. ヒアリング・アンケートの詳細	- 180 -
9-4-1. アジェンダ・タイムテーブル	- 180 -

9-4-2. アジェンダの詳細	- 180 -
9-4-3. 検証項目と評価方法	- 181 -
9-4-4. 実証実験の様子	- 182 -
9-5. 検証結果	- 185 -
10. 成果と課題	- 189 -
10-1. 本実証で得られた成果	- 189 -
10-1-1. 3D 都市モデルの技術面での優位性	- 189 -
10-1-2. 3D 都市モデルのビジネス面での優位性	- 190 -
10-2. 実証実験で得られた課題と対応策	- 191 -
10-3. 今後の展望	- 193 -
11. 用語集	- 194 -

1. ユースケースの概要

1-1. 現状と課題

都市部における建設工事では、建設工事の資材運搬等によって発生する交通渋滞の問題や、さまざまな大きさ・重量の資材を屋外・屋内へ移動させる「ラストワンマイル」輸送を効率化することの難しさが課題として挙げられている。これらの課題は、工事自体の遅延を引き起こすだけではなく、対象区画で通行止めの必要を生じさせるなど、近隣住民の生活や都市機能にも影響を及ぼす。

このため、自動搬送車両や自律運航型ドローン（無人ロボット）を資材運搬等に活用する方法が研究されているが、都市部における GPS による位置測位の不安定性や、SLAM 技術のコストの高さや精度の不足など、都市部における正確で安全な無人ロボット自動運航実現のための課題は多い。

上記課題の解決に向け、2022 年度の「[3D 都市モデルと BIM を活用したモビリティ自律運航システム](#)」及び 2023 年度の「[3D 都市モデルと BIM を活用したモビリティ自律運航システム（ドローン）v2.0](#)」「[3D 都市モデルと BIM を活用したモビリティ自動運航システム（車両）v2.0](#)」では、車両とドローンそれぞれにおいて以下の開発を行った。

車両においては、汎用的かつ高精度の自己位置測位手法として、3D 都市モデルを配置した仮想空間内の想定ルート上で仮想車両及び仮想 LiDAR を稼働させることで取得した点群から点群マップを作成するアルゴリズムを開発し、これを用いて現実空間の自己位置測位を行うシステムを開発した。

ドローンにおいては、3D 都市モデルと BIM を統合し点群化して得られた統合マップを用いた LiDAR 自己位置推定、GPS 測位、及びビジュアルイナーシャルオドメトリ（VIO）（連続的に取得されたカメラ画像の変化を捉えることでドローンの移動量を算出する技術）を用いた自己位置推定システムの結果を統合することで、高精度、低コスト、スケーラブルな屋内外でのドローンの自律運航システムを開発した。このシステムを用いた実証により、統合マップを用いた屋外から屋内へのドローンの自律運航が可能ことが確認され、一定の成果を得た。

一方で、自動搬送車両やドローンといった個々のモビリティの自動運航技術を開発したものの、ラストワンマイル輸送を社会実装するためには、水平・垂直輸送を統合した一貫した自動運航システムの開発によって相互の連携を図る必要がある。

1-2. 課題解決のアプローチ

今回の実証実験では、2023 年度の「[3D 都市モデルと BIM を活用したモビリティ自律運航システム（ドローン）v2.0](#)」「[3D 都市モデルと BIM を活用したモビリティ自律運航システム（車両）v2.0](#)」を基礎として開発したシステムに改修を加える。具体的には車両の自己位置推定機能及び車両・ドローンの自動運航計画立案/オペレーションが可能になるよう、ドローン用 GCS^{*1} アプリ「SENSYN CORE Pilot」（以下、SENSYN CORE Pilot）に Autocare（車両側の自動運転ソフトウェア）と連携するための MQTT^{*2} を活用したメッセージの双方向変換機能を加える。これにより、水平・垂直輸送を統合した自動運航システムを構築し、無人ロボット連携によるラストワンマイル輸送の実現を目指す。

本システムを用いることで、都市部の屋外空間や屋内空間、建築物に沿った垂直空間など、特徴の異なる空間をシームレスに移動するラストワンマイル輸送を行うための車両・ドローンの自動運航管理/モニタリングを簡易に実行できる仕組みを構築する。

*1 Ground Control System：無人航空機を人間が制御するためのシステム

*2 MQ Telemetry Transport：Message Queuing Telemetry Transport の略。メッセージ指向ミドルウェアのアプリケーション層で使用される、TCP/IP による Pub/Sub 型データ配信モデルの軽量なデータ配信プロトコル

1-3. 創出価値

昨今、都市部における建設工事では資材運搬等による交通渋滞が課題となっている。2 次元的な平面移動と比較して 3 次元的な移動が可能になると搬送経路の幅が広がり、交通渋滞の緩和が実現可能である。そのため自動運航可能なドローンや無人搬送車両（AGV）の活用による解決が期待されている。

一方で、GPS 測位のみでの自動運航では、ビルの間などにおいては自己位置の推定精度が悪くなり、正確で安全な運航を担保できないことがある。さらに、運航に必要な地図情報として民間事業者が提供する 3D マップを利用せざるを得ず、精度担保やデータ連携、カバレッジ等の点で課題がある。

そのため、GPS 測位による位置情報取得に加え、オープンデータである 3D 都市モデルを LiDAR 自己位置推定に利用することで、高精度で安定的な自己位置推定が行えることが期待されている。

また、現状は車両・ドローンは別個に自動運航を行うことが一般的であるため、自動運航管理/モニタリングシステムもそれぞれ存在し、複数のシステムを操作する必要がある。このような運用上の課題もあり、車両・ドローンを用いた水平・垂直方向の自動運航は「空と陸の新たなモビリティサービス」として社会実装されるに至っていない。

そのため、自動運航管理/モニタリングシステムを実現することで、新たなモビリティサービスとしての自動搬送車両及びドローンの社会実装を目指す。

1-4. 想定事業機会

表 1-1 想定事業機会

項目	内容
利用者	● 建設事業者 ● 運輸サービス事業者
サービス仮説	● 自動搬送車両走行及びドローン自律飛行によるラストワンマイル輸送サービス ➤ 単一システムで車両・ドローンの運航管理をワンストップで利用可能とすることで、事業者による無人ロボット自動運航の導入ハードルの低下や提供価値の向上に寄与できる
提供価値	● 3D 都市モデルを活用した、車両の自動走行及びドローンの自律飛行を可能とする運航管理/モニタリングシステムにより、都市部の建設工事において資材運搬等による交通渋滞の緩和を実現する

2. 実証実験の概要

2-1. 実証仮説

昨年度実証実験で改修を加えた SENSYN CORE Pilot に追加開発を加えることで、車両・ドローンを統合した自動運航計画立案ツール/自動運航オペレーショナルツールを構築する。車両・ドローンを同時に管理するインタフェースを備えることで、専門的な知識を持たないユーザーでも簡単に利用可能とする。

車両の自動運航において、従来の 3D 都市モデルによる自己位置推定に加え、RTK 測位データを活用することで、車両の自己位置推定精度を実用レベルに向上させる。

2-2. 実証フロー

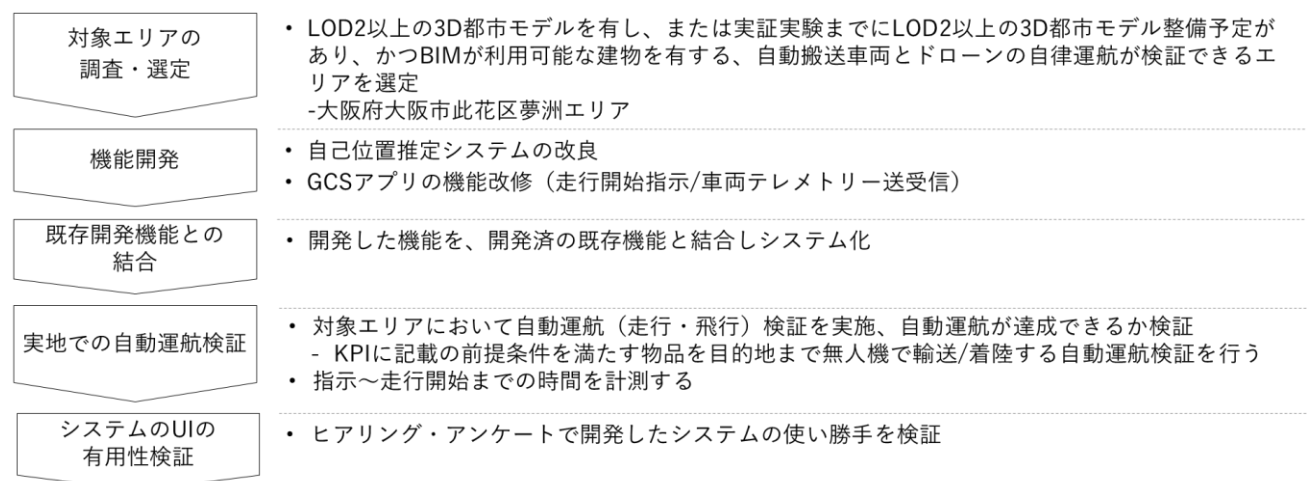


図 2-1 実証フロー

2-3. 検証ポイント

- 水平・垂直輸送の実現
 - SENSYN CORE Pilot の追加開発をすることで、従来は異なる手法や粒度で管理されていた車両・ドローンを統合し、同一 UI で管理可能とする。これにより、車両・ドローンのリアルタイムな位置情報や運航ステータスのモニタリングが単一のシステムで実施可能となる
 - 実際の資材運搬を想定し、1 サイズ 150g 程度の小型物品を積載し、水平・垂直方向の自動運航を複数回試行することで、輸送の実現性を確認する
- RTK 測位データを活用した車両の自己位置推定精度の向上
 - 昨年度実施した点群マッチングに加え、今年度では RTK 測位データで自己位置情報を補完する。RTK 測位データ補完の有無を比較することで、車両の自己位置推定精度を確認する

上記 2 点の検証ポイントについては、【7. 実証技術の機能要件の検証】で検証結果を記載

- 今回の実証実験で開発した車両・ドローンの自律運航計画立案ツールの有用性検証
 - 車両・ドローンの自動運航計画立案ツールとして、簡単に操作可能なユーザビリティの高いシステムになっているかを確認する
- 今回の実証実験で開発した車両・ドローンの自律運航オペレーショナルツールの有用性検証
 - 「SENSYN CORE Pilot」の追加開発により構築した今回のシステムが、車両・ドローンの自動運航をモニタリングする上で、簡単に操作可能なユーザビリティの高いシステムになっているかを確認する

上記 2 点の検証ポイントについては、【9. BtoB ビジネスでの有用性検証】で検証結果を記載

2-4. 実施体制

表 2-1 実施体制

役割	主体	詳細
全体管理	国交省 都市局	プロジェクト全体ディレクション
	アクセンチュア	プロジェクト全体マネジメント
実施事業者	竹中工務店	ユースケース実証におけるプロジェクト全体管理・運営 ユースケース実証において利用する BIM データの作成
	センシンロボティクス	ユースケース実証におけるドローン側システム構築の企画・開発・検証・運営
	アダワープジャパン	ユースケース実証における車両側システム構築の企画・開発・検証・運営
実施協力	2025 年日本国際博覧会協会	実証場所の提供 オープンデータ化の権利許諾

2-5. 実証エリア

表 2-2 実証エリア

項目	内容
実証地	大阪府大阪市此花区夢洲地区
面積	150,000 m ²
マップ (対象エリア は赤枠内)	 A map of the Yumoto area in Osaka, showing the experimental area highlighted in red. The map includes labels for '太陽光発電所' (Solar Power Generation Station), '夢洲中(一)' (Yumoto Nakamichi (1)), and '夢洲東(一)' (Yumoto Higashi (1)). A red rectangle indicates the experimental area. A black crosshair is also visible on the map. The map shows a coastal area with a river and a bridge labeled '夢洲大橋' (Yumoto Bridge). A tunnel labeled '夢洲トンネル' (Yumoto Tunnel) is also shown.

2-6. スケジュール

表 2-3 スケジュール

実施事項	令和 6 年										令和 7 年		
	4 月	5 月	6 月	7 月	8 月	9 月	10 月	11 月	12 月	1 月	2 月	3 月	
1. 水平・垂直方向の自動運航に関する要件定義・実証方法の検討													
2. BIM・3D 都市モデルの整備													
3. 車両ソフトウェア開発													
4. 車両連携システム開発													
5. 車両・ドローン統合ユーザーインターフェース開発													
6. 実走行・飛行検証													
7. ヒアリング・アンケート実施													
8. 成果取りまとめ													

3. 開発スコープ

3-1. 概要

今回の実証実験では、ラストワンマイル輸送に必要な「水平・垂直方向の自動運航」の実現を目的として、車両・ドローンの自動運航計画立案/オペレーションツールの開発を行う。

また、車両の自己位置推定能力改善のため、点群マッチングだけでなく、RTK 測位データで自己位置情報を補完する仕組みを構築する。

3-2. 開発内容

本プロジェクトでは、建設現場における建築資材のラストワンマイル輸送の効率化を目指し、ドローンと自動搬送車両の双方に対応した自動運航システムを開発した。基盤システムとしては国内で開発されたブラウザベースのドローン用 GCS アプリ SENSYN CORE Pilot を選定し、SENSYN CORE Pilot に改修を加えることで、同一システム上でドローンおよび車両のルート作成およびモニタリングできる環境を構築した。また、車両の自己位置推定精度を向上させるため、自動搬送車両に RTK-GNSS 測位デバイスを搭載し、従来の点群マップを用いた自己位置推定と組み合わせた新たな自己位置推定アルゴリズムを開発した。

SENSYN CORE Pilot は、ドローンの飛行ルートの設定と、飛行中のドローン位置の表示や飛行指示を行うためのオペレーティングツールであり、直感的なドローン運航を目的として利用している。本システムはこれまでに、自己位置推定システムから取得した自己位置測位情報を SENSYN CORE Pilot 上でリアルタイムに表示しオペレーターの飛行操作を支援するため、地形データ切替え機能、PLATEAU-Terrain 対応機能、ROS メッセージ双方向変換機能、非常停止機能を開発・搭載した。SENSYN CORE Pilot とドローンの双方向の通信システムを確立している。

Autoware は、自動搬送車両の開発と運用を支援するためのオペレーションシステムであり、スムーズな自動運転の実現を目指して利用されている。本システムは、車両のセンサーから取得したデータをリアルタイムに処理・表示し、オペレーターの運行操作を支援する機能を備えている。具体的には、自己位置推定機能、障害物検知と回避機能、経路計画アルゴリズム、ROS メッセージ双方向変換機能、緊急停止機能が開発・搭載されている。

本プロジェクトではドローンと車両を同一システムで運航管理するための UI を新たに開発した。また、本来ドローンの自律運航/モニタリングシステムである SENSYN CORE Pilot と、車両の自動運行オペレーティングシステムである Autoware の間で情報を双方向にやり取りするため、MQTT*2 を活用して Autoware からの情報を適切な形式に変換する仕組みを構築した。これにより、双方のシステム間でデータの受け渡しが可能と

なり、SENSYN CORE Pilot 上で車両・ドローンのテレメトリー情報や、自律運航の開始や完了の確認が可能となった。

次に、車両の自己位置推定精度向上のため、Autoware の自己位置推定機能に、RTK-GNSS 測位のデータを取り込み、従来の点群マッチングの自己位置推定精度が低下した場合に RTK-GNSS 測位の自己位置に置き換えるロジックを新たに追加した。具体的には、自動搬送車両のハードウェア構成として、電子制御可能な 4 トン電動トラックにソフトバンク社製の RTK-GNSS 測位デバイス「ichimill」と Ouster 社製の LiDAR「OS-1-64」、自己位置測位のソフトウェアを実行する PC は System76 社製「kudo」を使用し、「ichimill」から取り込んだ NMEA 形式のデータを LiDAR「OS-1-64」から取得した点群データと併用した。従来は点群データを活用した自己位置推定のみ行っていたが、これに加えて、点群データによる自己位置推定精度（点群マッチングの相関値）が一定の基準を下回った場合に、RTK-GNSS 測位で自己位置推定を補うことで精度向上を図った。

ドローンのハードウェア構成としては、2023 年度の「3D 都市モデルと BIM を活用したモビリティ自律運航システム（ドローン）v2.0（<https://www.mlit.go.jp/plateau/use-case/uc23-17-1/>）」同様、DJI 社製「DJI Matrice 300 RTK」をベースとした独自機体を使用した。

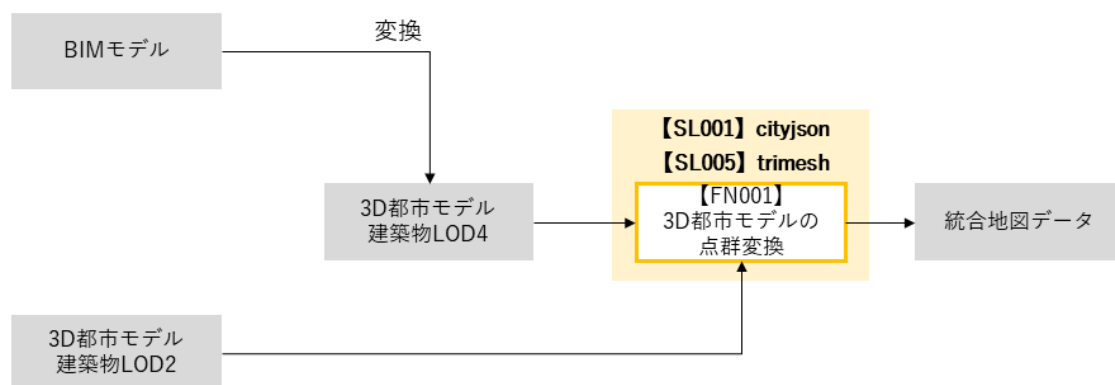
実証エリアとしては、大阪府大阪市夢洲地区の施設整備事業（PW 西工区）の工事現場を選定した。自律運航に必要な 3D マップ生成のため、3D 都市モデルの建築物モデル（LOD3）および都市設備モデル（LOD3）をベースに一部建築物の BIM モデルを変換して得られた建築物モデル（LOD4）をマージすることで、一つの CityGML ファイルを生成した。次に、ドローン用の点群マップについては、Trimesh ライブラリを用いてこの CityGML ファイルの表面をサンプリングすることで生成し、自動搬送車両用の点群マップとしては、PLATEAU PCL Generator 経由で生成された点群マップ「PCLG マップ」を利用した。

本システムの検証として、ドローン・車両のルート作成・運航管理の有用性、車両の自己位置推定精度を検証した。

4. 実証システム

4-1. アーキテクチャ

4-1-1. システムアーキテクチャ



凡例

既存のソフトウェア	開発したソフトウェア	既存機能	開発した機能	データ	ファイルストレージ	データベース
-----------	------------	------	--------	-----	-----------	--------

図 4-1 システムアーキテクチャ①（ドローン用統合地図データ生成）

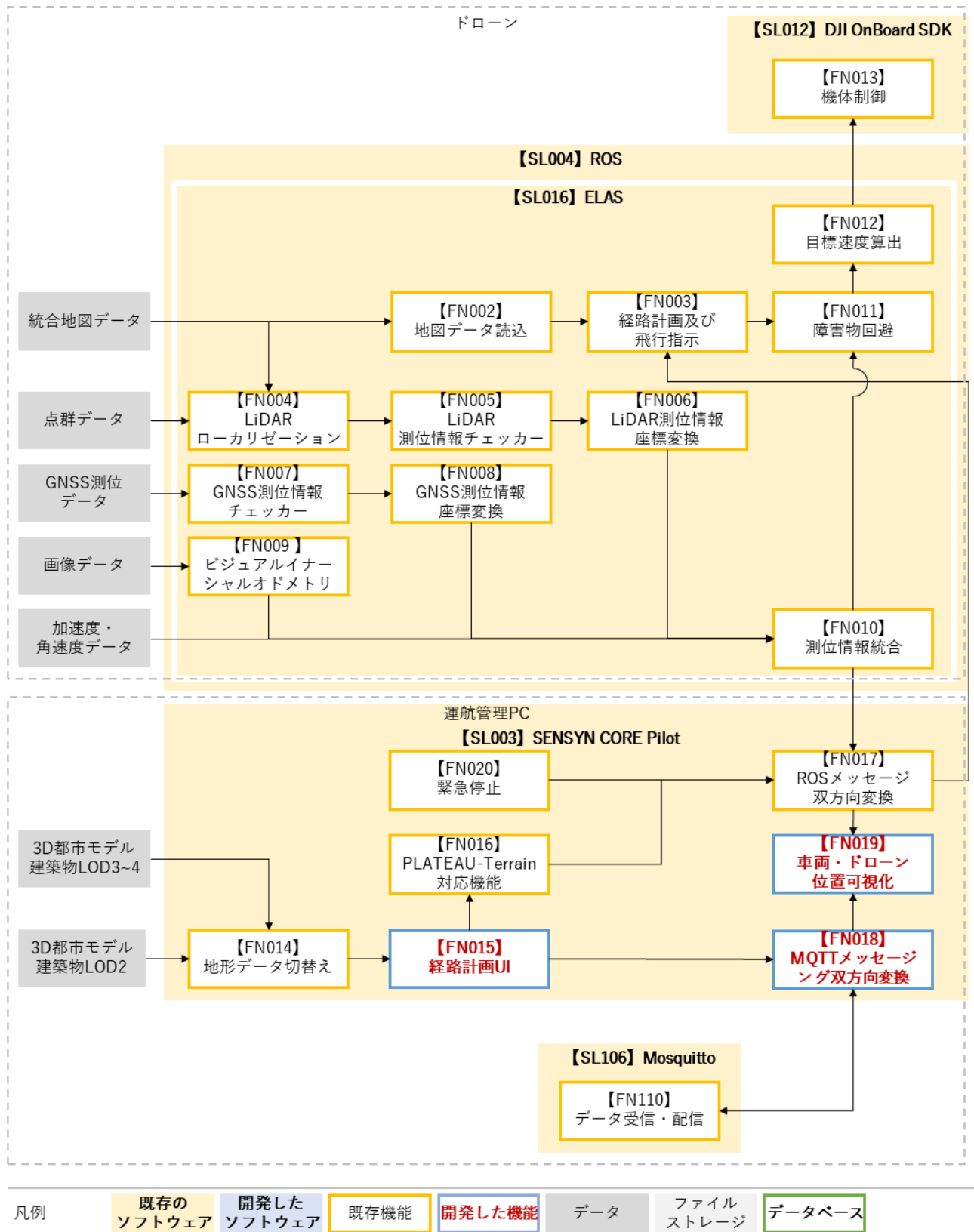


図 4-2 システムアーキテクチャ② (飛行計画～ドローン自律運航)

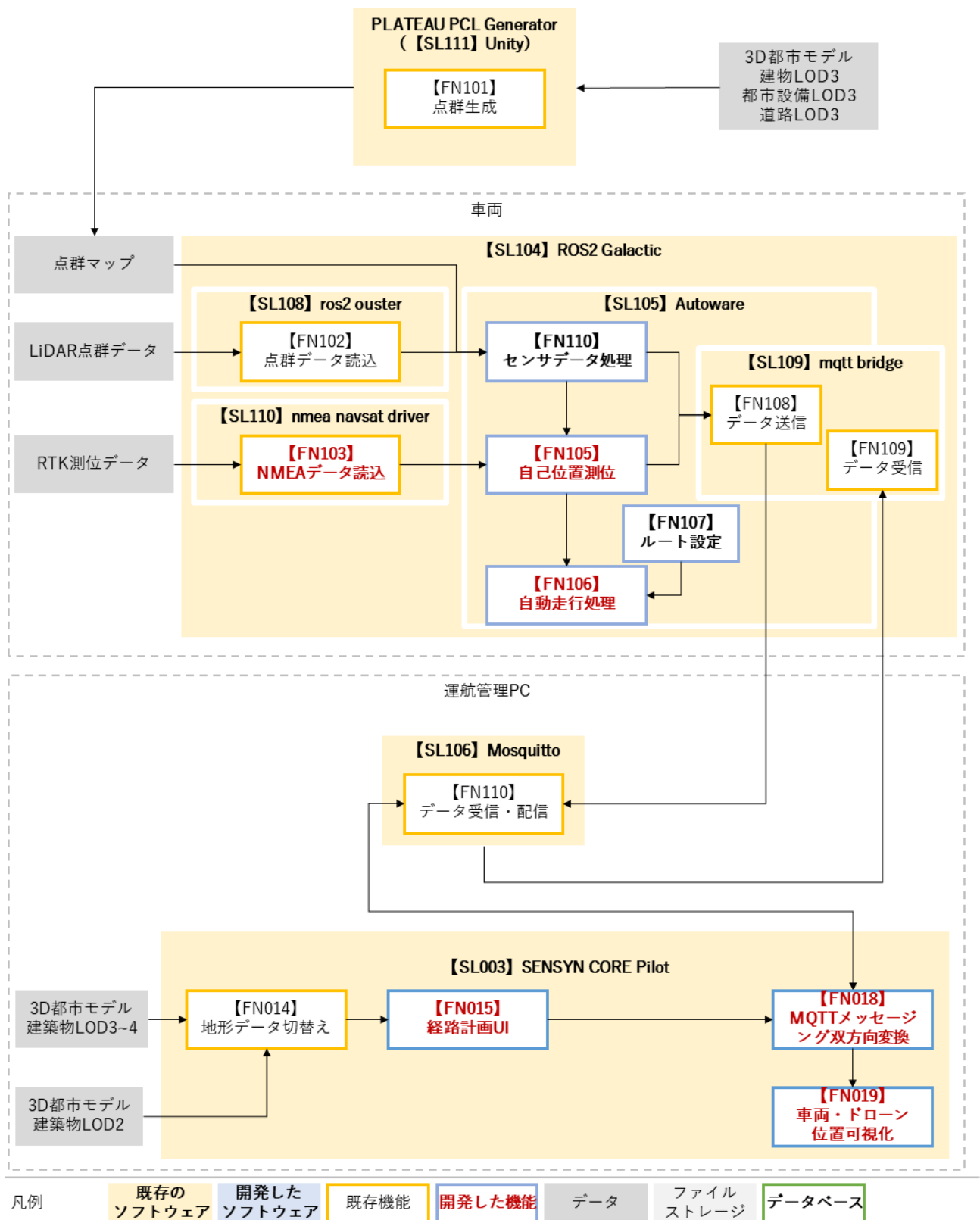
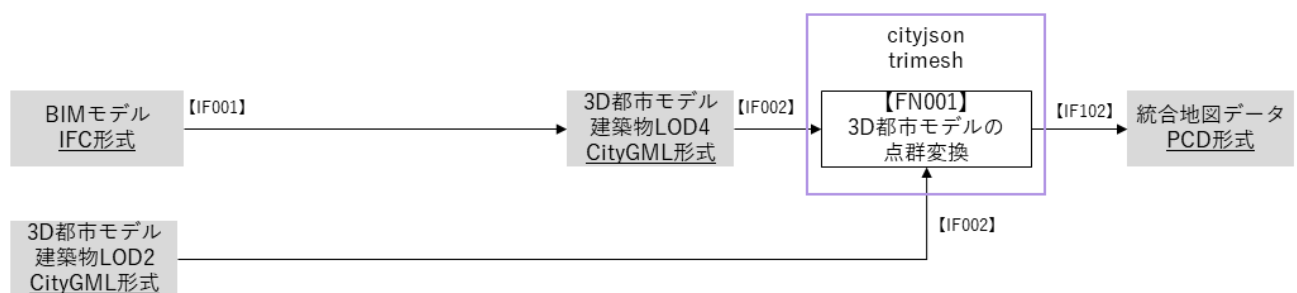


図 4-3 システムアーキテクチャ③（車両用点群マップ作成～車両自動運転）

4-1-2. データアーキテクチャ



凡例

UI入力

データ

データベース

ファイル
ストレージ

ソフトウェア

データ
処理

図 4-4 データアーキテクチャ①（統合地図データ生成）

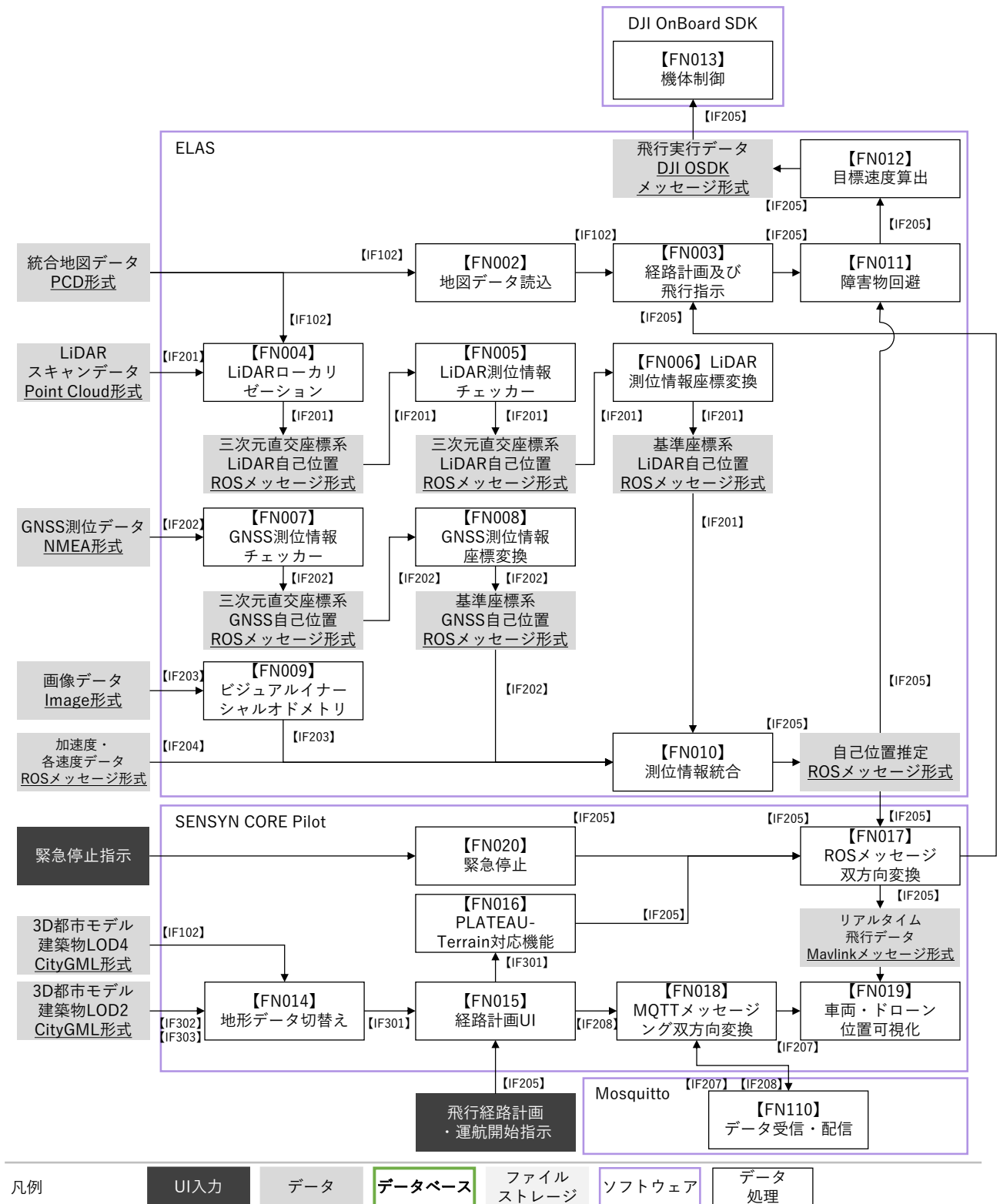


図 4-5 データアーキテクチャ②（飛行計画～ドローン自律運航）

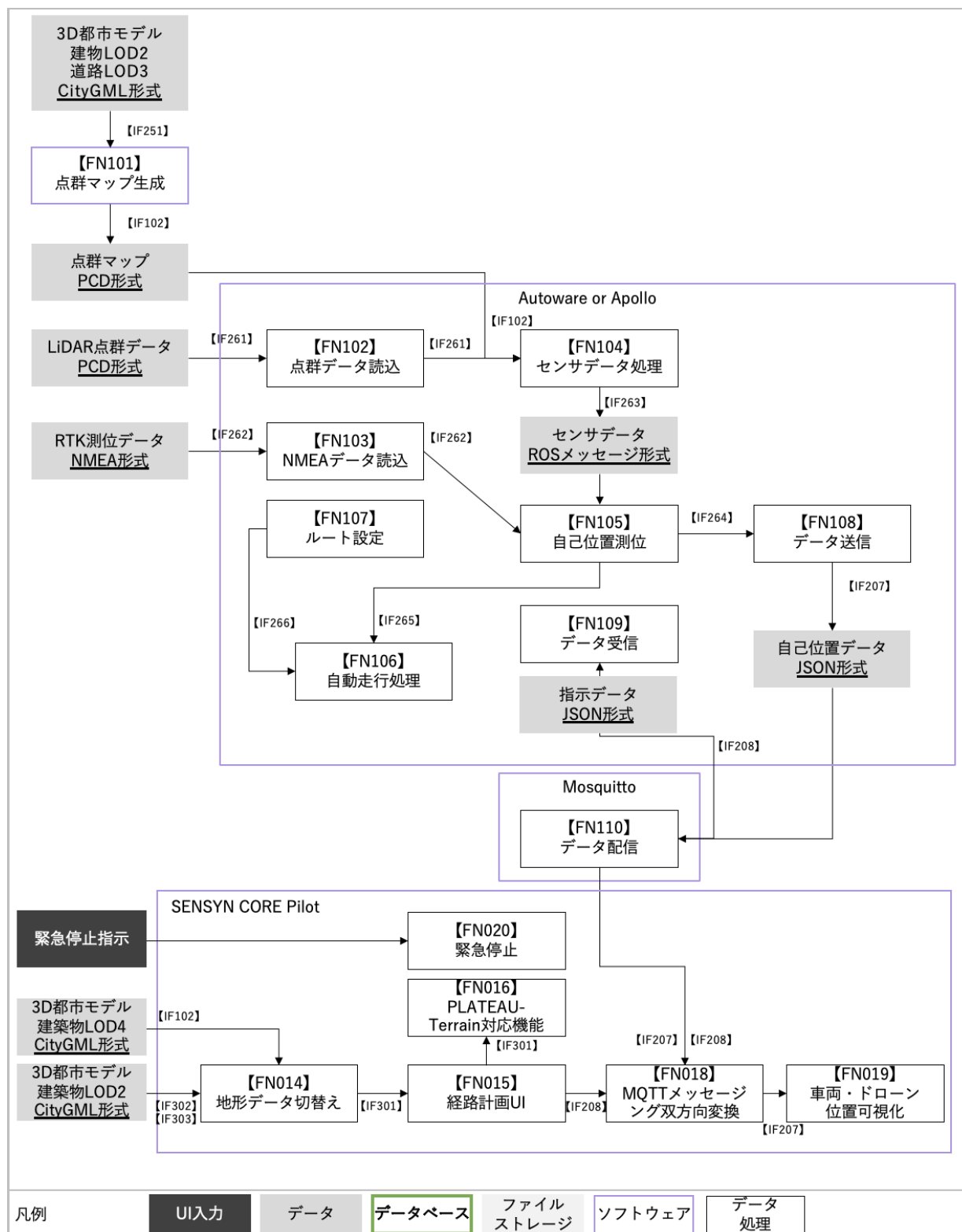


図 4-6 データアーキテクチャ③（車両用点群マップ作成～車両自動運転）

4-1-3. ハードウェアアーキテクチャ

4-1-3-1. 利用したハードウェア一覧

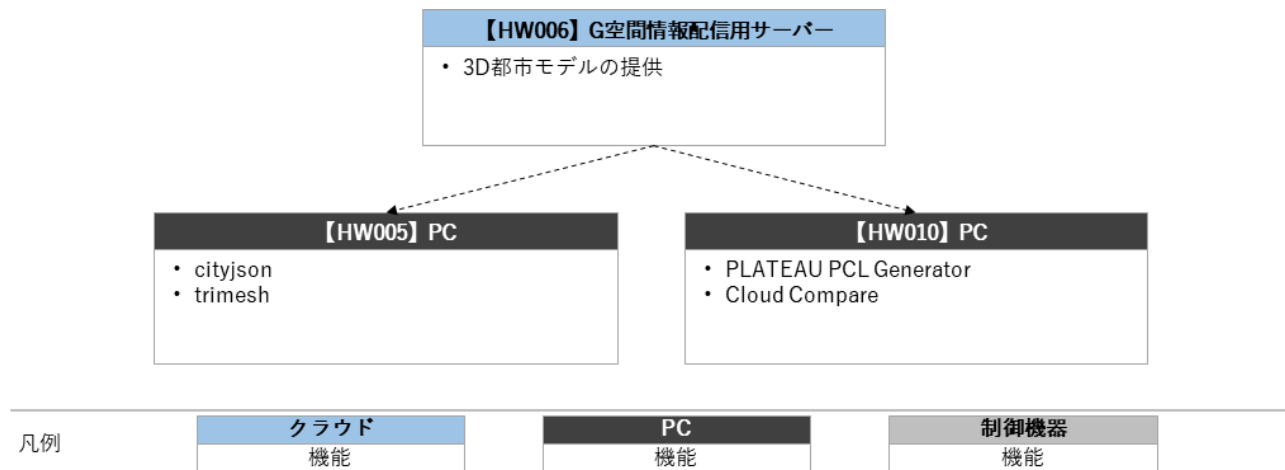


図 4-7 ハードウェアアーキテクチャ①（点群マップ生成）

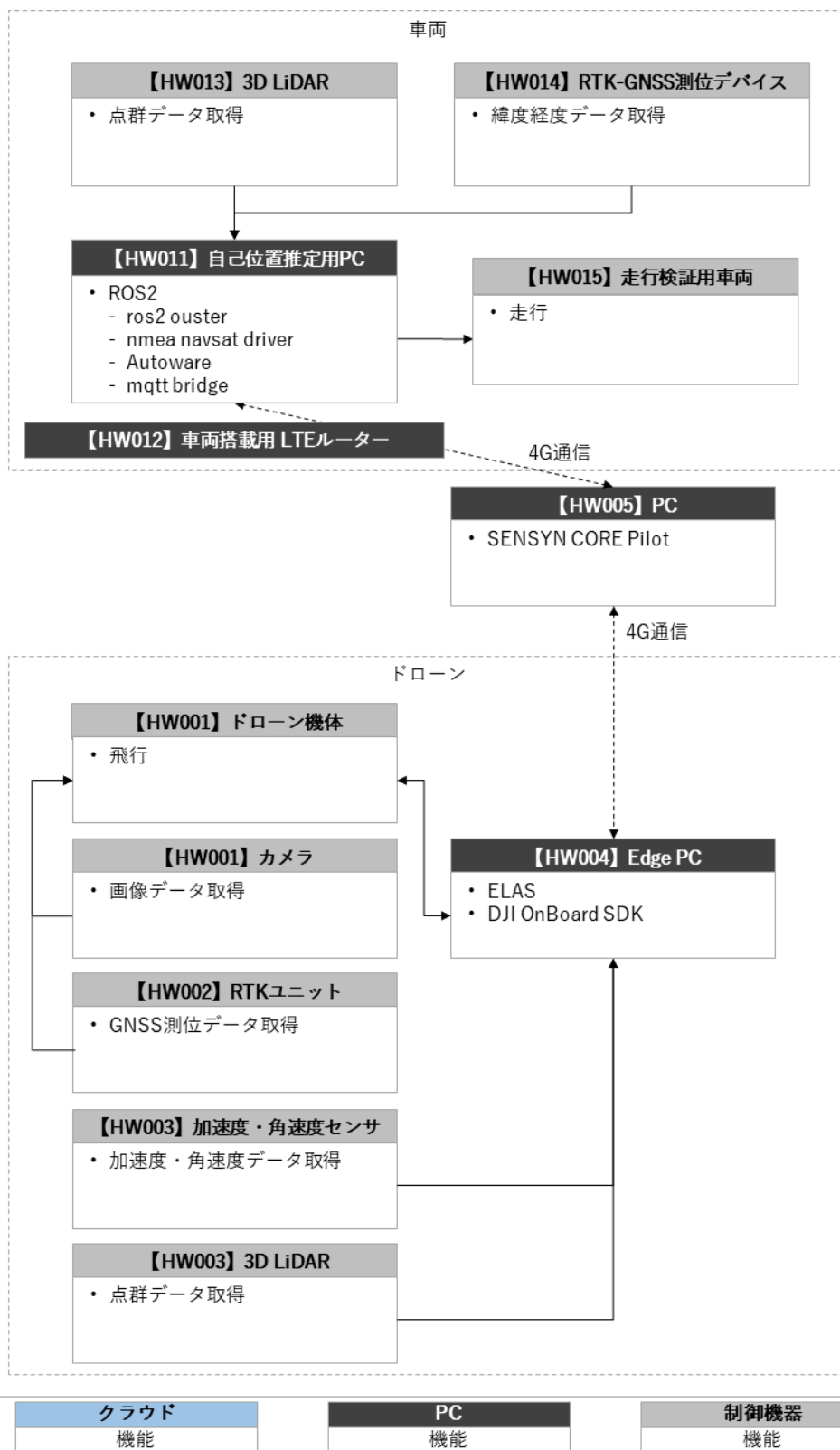


図 4-8 ハードウェアアーキテクチャ②（飛行計画～ドローン自律運転）

表 4-1 利用したハードウェア一覧

ID	種別	品番	用途
HW001	ドローン	DJI Matrice 300	● 3D LiDAR 及び Edge PC を搭載したドローン飛行のために利用
HW002	RTK ユニット	HW001 に標準搭載されたものを使用	● GNSS 測位による自己位置推定（ドローン用）のために利用
HW003	3D LiDAR	Ouster OS0-64	● LiDAR センサーを用いた自己位置推定（ローカリゼーション）のための点群取得（ドローン用）に利用
HW004	Edge PC	PICO-WHU4W2-A11-0004	● 各センサー情報を処理し、LiDAR ローカリゼーションやビジュアルイナercialオドメトリ（VIO）、GNSS 測位を用いた FUSION のために利用
HW005	PC	Lenovo 21B4S0C700	● 点群マップの生成、及び Web ブラウザ上で SENSYN CORE Pilot を使用するために利用
HW006	G 空間情報配信サーバー	-	● 点群マップ生成に必要なデータを配信する唯一のサーバー
HW010	点群マップ生成用 PC	GALLERIA XL7C-R45	● PLATEAU PCL Generator を起動し、PC モニターで点群データを確認するために利用
HW011	自己位置推定用 PC	System76 Kudu	● 自己位置測位のソフトウェアを動かすための Ubuntu PC
HW012	車両搭載用 LTE ルーター	GLiNet Puli	● 外部システム連携のために車両に搭載する LTE ルーター
HW013	3D LiDAR	Ouster OS-1-64	● 自己位置測位の点群マッチング（車両用）のために用いる 3D LiDAR
HW014	RTK-GNSS 測位デバイス	ichimill	● 自己位置測位の点群マッチング（車両用）のために用いる RTK-GNSS 測位デバイス
HW015	走行検証用車両	EPT-4T	● 搬送車両の車体に用いる 4 トン電動トラック

4-1-3-2. 利用したハードウェア詳細

1) 【HW001】 ドローン：DJI Matrice 300

- 選定理由
 - Edge PC と LiDAR を搭載しやすい設計
 - 外部 SDK を具備
 - ペイロードが十分ある
- 仕様・スペック
 - サイズ:810×670×430mm
 - 最大積載量:2.7 kg
 - 最大風圧抵抗:15 m/秒
 - 最大飛行時間:55 分
- イメージ



図 4-9 DJI Matrice 300¹

¹ 公式 HP より抜粋：<https://enterprise.dji.com/jp/matrice-300>

2) 【HW002】 RTK ユニット：HW001 に標準搭載されたもの

- 選定理由
 - 機体にインテグレーション済み
- 仕様・スペック
 - 対応する衛星測位システム：GPS+GLONASS+BeiDou+Galileo
 - RTK 測位精度：1cm+1ppm（水平方向）、1.5cm+1ppm（垂直方向）

3) 【HW003】 3D LiDAR：Ouster OS0-64

- 選定理由
 - 超広視野角
 - 軽量（445g）
- 仕様・スペック
 - 反射率 10% でスキャン距離 30m 以上
 - 最大測定範囲 50m 以上
 - 垂直視野 90°
 - ビーム本数 64 本の解像度
 - 520 万ポイント/秒
- イメージ



図 4-10 Ouster OS0-64²

² 公式 HP より抜粋：<https://ouster.com/jp/products/scanning-lidar/os0-sensor/>

4) 【HW004】 Edge PC : PICO-WHU4W2-A11-0004

- 選定理由
 - 筐体が小型であり、各センサー情報を処理するのに十分な CPU 処理性能を持つこと
- 仕様・スペック
 - サイズ：縦 100mm×横 72mm
 - CPU：k[®] Core™ i7-8665UE
 - Storage：256GB
- イメージ



図 4-11 PICO-WHU4W2-A11-0004（オモテ面）



図 4-12 PICO-WHU4W2-A11-0004（ウラ面）³

5) 【HW005】 PC : Lenovo 21B4S0C700

- 選定理由
 - SENSYN CORE Pilot アプリを利用するのに十分なスペックである
- 仕様・スペック
 - サイズ: 約 305×218×17.3mm
 - CPU : Intel®i7-1255U 1.70 GHz
 - OS : Windows 10 Business 64bit
 - メモリ : 16GB
 - ストレージ : 512GB
- イメージ



図 4-13 Lenovo 21B4S0C700⁴

⁴ 公式 HP より抜粋 : [https://www.lenovo.com/jp/ja/p/laptops/thinkpad/thinkpad-l-series/thinkpad-l13-gen-3-\(13-inch-intel\)/len101t0031?orgRef=https%253A%252F%252Fwww.google.com%252F](https://www.lenovo.com/jp/ja/p/laptops/thinkpad/thinkpad-l-series/thinkpad-l13-gen-3-(13-inch-intel)/len101t0031?orgRef=https%253A%252F%252Fwww.google.com%252F)

6) 【HW006】 G 空間情報配信用サーバー

- 選定理由
 - 点群マップ生成に必要なデータを配信する唯一のサーバーであるため
- 仕様・スペック
 - <https://front.geospatial.jp/>

7) 【HW010】 点群マップ生成用 PC : GALLERIA XL7C-R45

- 選定理由
 - 点群マップ作成処理に必要なメモリを持つため
 - 同様のスペックを持つ PC が監視用、遠隔操作で合わせて 2 台必要なため
- 仕様・スペック
 - サイズ：縦 245mm×横 360mm×高さ（最薄部）20mm
 - OS：Windows 11
 - CPU：Intel® Core™ i7-13700H
 - メモリ：16GB
 - ストレージ：500GB
- イメージ



図 4-14 GALLERIA XL7C-R45

8) 【HW011】 自己位置推定用 PC : System76 Kudu

- 選定理由
 - 自己位置測位のソフトウェアを動かすために Ubuntu PC が必要なため
- 仕様・スペック
 - サイズ：縦 258mm×横 361mm×高さ 29mm
 - OS：Ubuntu 20.04
 - CPU：AMD Ryzen™ 5900HX
 - メモリ：16GB
 - ストレージ：1TB
- イメージ



図 4-15 System76 Kudu

9) 【HW012】車両搭載用 LTE ルーター：GLiNet Puli

- 選定理由
 - 今回の実証実験に十分な通信速度が担保可能であり、充電式バッテリー付属のため実証が長時間になった場合にも対応可能なため
- 仕様・スペック
 - サイズ：縦 120mm×横 74mm×高さ 28mm
 - CPU：QCA9531 @650MHz SoC
 - Wi-Fi 速度：300Mbps(2.4GHz)
 - Storage：NOR フラッシュ 16MB + NAND フラッシュ 128MB
- イメージ



図 4-16 GLiNet Puli

10) 【HW013】 3D LiDAR : Ouster OS1-64

- 選定理由
 - 超広視野角
 - 軽量 (445g)
- 仕様・スペック
 - 最大測定距離 200m
 - 垂直視野 45 度
 - ビーム本数 64 本の解像度
 - 2,621,440 ポイント/秒
- イメージ



図 4-17 3D LiDAR : Ouster OS1-64

11) 【HW014】 RTK-GNSS 測位デバイス：ichimill

- 選定理由
 - 点群マップ作成に必要なため
 - 従来の衛星測位に比べ、誤差が数 cm に抑えられるため
- 仕様・スペック
 - リアルタイムで誤差数 cm の測位が可能
 - 準天頂衛星 GNSS から GNSS 信号を受信して測位
- イメージ



図 4-18 ichimill

12) 【HW015】 走行検証用車両

- 選定理由
 - 電子制御可能なため
 - 4 トンの重量を積載可能なため
- 仕様・スペック
 - 積載可能重量：4000kg
 - 重量：2600kg
- イメージ



図 4-19 自動搬送車両

4-2. システム機能

4-2-1. システム機能一覧

1) ドローン機能一覧

表 4-2 ドローン用機能一覧

※赤文字：新規開発・既存改修

大分類	小分類	ID	機能名	機能説明
統合地図データの作成	CityGML to Robot Map	FN001	3D 都市モデルの点群変換	● 3D 都市モデルから点群 (Point Cloud) を生成する
自己位置測位	ELAS	FN002	地図データ読込	● 自己位置推定用の地図として利用するために点群マップを読み込む
		FN003	経路計画及び飛行指示 ・ウェイポイント (基準座標系) ・緊急停止指示	● ユーザーが入力した経路計画 (ウェイポイント/アクション) を受け取る
		FN004	LiDAR ローカリゼーション	● ドローンに搭載した 3D LiDAR で取得したデータと 3D 都市モデルから変換した点群マップを重ね、自己位置を推定する
		FN005	LiDAR 測位情報チェッカー	● 3D LiDAR による測位情報の信頼性評価を行う機能
		FN006	LiDAR 測位情報座標変換	● 3D LiDAR による測位情報を基準座標系へ変換する機能
		FN007	GNSS 測位情報チェッカー	● GNSS による測位情報の信頼性評価を行う機能
		FN008	GNSS 測位情報座標変換	● GNSS による測位情報を基準座標系へ変換する機能
		FN009	ビジュアルイナercialオドメトリ (VIO)	● ドローン内蔵カメラで撮影した、連続した画像の間に起こる変化、変移を捉えることで算出した移動体の移動量に加え、IMU 情報を加味して移動量を推定する機能
		FN010	測位情報統合	● 複数の測位情報を統合する機能
		FN011	障害物回避	● 障害物が存在する場合は障害物を避ける経路を生成し、移動目標点を更新する機能
		FN012	目標速度算出	● 自己位置と目標位置からドローンの移動速

ドローン制御 ソフトウェア SENSYN CORE Pilot	経路計画			度を決定/指示する機能
		FN013	機体制御	● 飛行指示に従い機体を制御する機能
		FN014	地形データ切替え	● 3D 都市モデルと BIM の統合マップを SENSYN CORE Pilot 上に表示するために地形データの切替えを行う機能
		FN015	経路計画 UI	● 車両・ドローンの経路計画を行う機能
	HW 接続	FN016	PLATEAU-Terrain 対応機能	● SENSYN CORE Pilot を用いて経路計画を行う際に、SENSYN CORE Pilot で用いられている標高データと異なる標高データ（PLATEAU-Terrain）を用いた場合、ドローンへのルート転送時に標高差を補正する機能
		FN017	ROS メッセージ双方向変換	● SENSYN CORE Pilot と ROS により動作するロボットの間で、情報を双方向にやり取りするのに適切な形式に変換する機能
		FN018	MQTT メッセージ双方向変換	● SENSYN CORE Pilot と車両の間で、情報を双方向にやり取りするため、MQTT メッセージを json 形式に変換する機能
		FN019	位置可視化	● 車両・ドローンの現在位置をリアルタイムで表示する機能
	HW 制御	FN020	緊急停止	● ドローンを緊急停止させる指令を ELAS に送る機能

2) 車両用機能一覧

表 4-3 車両用機能一覧

※赤文字：新規開発・既存改修

大分類	小分類	ID	機能名	機能説明
基本機能	点群マップ生成	FN101	点群生成	● 3D 都市モデルから仮想 LiDAR を用いて点群マップを出力する（PLATEAU PCL Generator を利用）機能
	車両自動運転	FN102	点群データ読込	● LiDAR の点群データを取り込む機能
		FN103	NMEA データ読込	● RTK 測位デバイスから NMEA 形式のデータを取り込む機能
		FN104	センサーデータ処理	● 車両に搭載した LiDAR、RTK 測位、IMU からセンサーデータの情報を取得と処理を行

				う機能
		FN105	自己位置測位	● 点群マッチング、RTK 測位データを基に自己位置測位処理を行う機能
		FN106	自動走行処理	● Autoware のルート設定機能から送信される車両の動作指示又は SENSYN CORE Pilot から送信される車両の停止指示を受け取り、車両への動作命令（走行、ステアリング、ブレーキ処理）を出力する機能
		FN107	ルート設定	● 仮想空間内に仮想車両を走行させるために、ウェイポイントを設定する機能
		FN108	データ送信	● 車両の測位データを SENSYN CORE Pilot へ送信する機能
		FN109	データ受信	● SENSYN CORE Pilot から送られる車両停止指示データや走行ルートを受信する機能
	遠隔監視	FN110	データ受信・配信	● 車両と SENSYN CORE Pilot 間のデータ受信・配信を MQTT メッセージで送受信するためにサーバーに設置される機能

4-2-2. 利用したソフトウェア・ライブラリ

表 4-4 ドローンで利用するソフトウェア・ライブラリ

ID	項目	内容
SL001	cityjson	● オープンソースとして提供されているデータフォーマット ● CityGML 形式の 3D 都市モデルを OBJ データに変換
SL002	Ubuntu 24.04 LTS	● Canonical 社が提供する Linux Distribution
SL003	SENSYN CORE Pilot	● センシンロボティクス社の自社製 GCS アプリ。無人航空機の経路計画、制御、機体位置の可視化などを行う
SL004	ROS	● Open Robotics が提供するロボットオペレーションシステム
SL005	trimesh	● オープンソースとして提供されている Python ライブラリ ● 3D オブジェクトを多角形の集合で表現し、三角メッシュの読み込み操作・分析が可能
SL006	Point cloud library	● オープンソースとして提供されている 3 次元点群処理が可能なライブラリ
SL007	proj	● オープンソースとして提供されている複数の測地系を扱うことを可能とするライブラリ
SL008	OpenCV	● オープンソースとして提供されている画像処理アルゴリズムなど

		の機能を提供するライブラリ
SL009	Octomap	● オープンソースとして提供されているセンサー情報から 3D 環境認識ができるライブラリ
SL010	ompl	● Kavraki Lab (Rice University) が提供する Open Motion Planning Library ● 経路計画アルゴリズムのライブラリ
SL011	Boost	● オープンソースとして提供されている C++ の標準ライブラリ
SL012	DJI OnBoard SDK	● DJI 社から提供されているドローン内のハードウェア制御のための開発ツール ● ドローンに搭載したシリアルポート(TTLUART) を使用して、オンボードコンピューターに対応するフライトコントローラーに接続可能
SL013	Cesium Geocode API	● Cesium 社が提供するジオコード API ● ジオコードとは、住所情報を数値形式のコードに変換したもの
SL014	egm96-universal	● アメリカ国家地球空間情報局 (National Geospatial-Intelligence Agency、NGA) が提供する EGM96 のジオイドモデルデータからジオイド高を取得するライブラリ
SL015	MAVLink toolchain	● MAVLink はオープンソースとして提供されており、ドローンと GCS の間で機体の状態やルート情報などのデータを送受信するメッセージフォーマット ● MAVLink toolchain によりカスタムメッセージフォーマットを XML で定義することで C++ や Python のヘッダファイルを自動的に生成する
SL016	ELAS	● センシンロボティクス社が提供する Edge LiDAR Autonomous System の略称 ● ロボットの自律移動に必要な、認識、計画、制御機能を有したソフトウェアパッケージ ● 事前マップを用いて自己位置測位、かつ経路計画を行い、ロボットに移動経路/指示を与えて、自律移動を行う技術 ● 「事前マップ」について、今回の実証実験では 3D 都市モデルを点群化したデータを事前マップとして活用することで、SLAM による事前マップ作成手法と比較して低コスト、高精度、スケーラブルなマップ作成を実現している ● 「自己位置測位」について、具体的には LiDAR と GNSS (衛星測位システム) による自己位置推定に加え、ビジュアルイナージアルオドメトリ (VIO) による自己位置測位を利用

表 4-5 車両で利用するソフトウェア・ライブラリ

ID	項目	内容
SL101	Ubuntu 20.04 LTE	● Autoware で自己位置推定を用いるための開発用 PC にインストールされている OS
SL102	Windows 11	● デジタルツインビュー、遠隔映像監視で用いる開発用 PC にインストールされている OS
SL103	PLATEAU PCL Generator	● 事前取得した都市モデルデータを基に、仮想空間内で仮想車両を走行させ、マップを取得するソフトウェア
SL104	ROS2 Galactic	● ロボットアプリケーション開発のためのミドルウェア
SL105	Autoware 1.11.0	● オープンソースの自動運転ソフトウェアで、センサーデータの処理や制御アルゴリズムを提供し、自動車の自律的な走行を実現するためのプラットフォーム
SL106	Mosquitto 5.0	● パブリッシュ/サブスクライブ(Publish/Subscribe)型のメッセージングプロトコルである「MQTT」における、ブローカー（サーバー側）のミドルウェア
SL107	PLATEAU Unity SDK for Unity 3.1.1	● 3D 都市モデルデータを Unity で扱うためのライブラリ
SL108	ROS2-ouster	● Ouster LiDAR を Autoware で読み込めるようにするための ROS2 ライブラリ
SL109	Mqtt_bridge	● Autoware から自己位置情報や車両情報を MQTT Broker へ送るための ROS2 ライブラリ
SL110	nmea navsat driver	● RTK-GNSS 測位デバイスから得られる NMEA 形式のデータを Autoware で読み込めるようにするための ROS2 ライブラリ
SL111	Unity 2022.3.25f1	● ゲームやアプリを開発するための統合開発 環境
SL112	Autoware-sensing	● 3D LiDAR や GNSS、IMU のデータを受け 取り、Autoware 内で利用するための処理を 行うライブラリ。本システムでは 3D LiDAR のデータのみを利用
SL113	Autoware-localization	● Autoware-sensing から受け取ったセンサーデータ、また点群マップを用いて自己位置測位を行うライブラリ
SL114	lidar slam_ros2 0.1.0	● NDT スキャンマッチングを利用した ROS2 の SLAM 用ライブラリ

4-2-3. 開発機能の詳細要件

開発機能の詳細要件を記す。なお、今回の実証実験において新規に開発する要素（機能名）を赤字で示す。

1) ドローン用機能一覧

1. 【FN001】3D 都市モデルの点群変換

- 機能概要
 - 3D 都市モデルから点群（Point Cloud）を生成する
- フローチャート

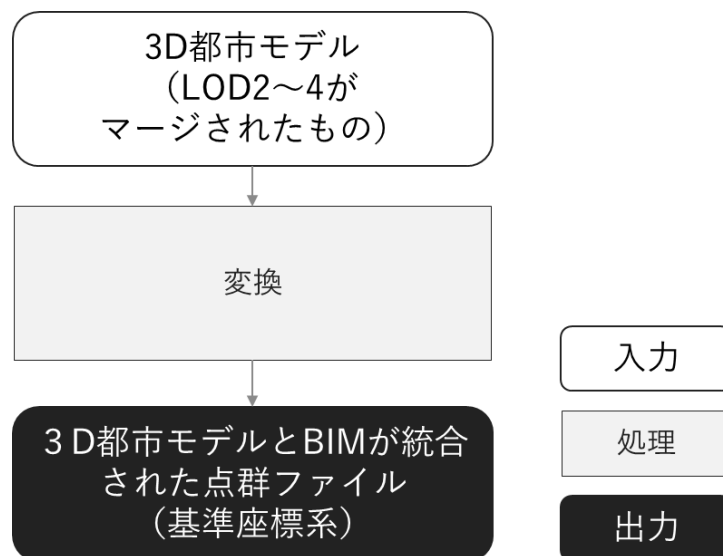


図 4-20 3D 都市モデルの点群変換機能のフローチャート

- データ仕様
 - 入力
 - ◇ 3D 都市モデル（LOD2～4 がマージされたもの）
 - 内容
 - 対象エリアの BIM モデルから作成した LOD4 データをマージした 3D 都市モデル（LOD2～LOD4）
 - 形式
 - CityGML 形式
 - データ詳細
 - ファイル入力インタフェース【IF001】を参照
 - 出力
 - ◇ 3D 都市モデルと BIM が統合された点群ファイル（基準座標系）
 - 内容
 - 利用した 3D 都市モデル【DT001】【DT002】【DT004】を参照

- 利用した BIM モデル【DT101】を参照
- 形式
 - Point Cloud 形式
- データ詳細
 - ファイル出力インターフェース【IF101】を参照
- 機能詳細
 - CityGML 形式の 3D 都市モデルマップからドローン飛行用の点群 (Point Cloud)マップを生成
 - ✧ 処理内容
 - CityGML 形式のマップを obj ファイルに変換したのち点群に変換する
 - ✧ 利用するライブラリ
 - cityjson (ソフトウェア・ライブラリ【SL001】を参照)
 - trimesh (ソフトウェア・ライブラリ【SL005】を参照)
 - ✧ 利用するアルゴリズム
 - なし

2. 【FN002】地図データ読み込み

- 機能概要
 - ドローン用の点群マップを ELAS に読み込む
- フローチャート

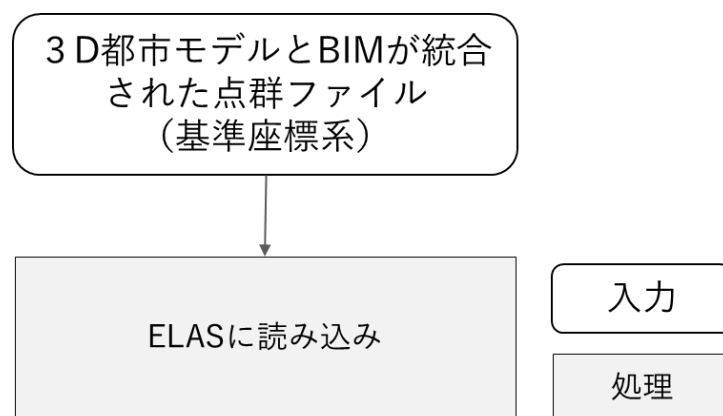


図 4-21 地図データ読み込み機能のフローチャート

- データ仕様
 - 入力
 - ✧ 3D 都市モデルと BIM が統合された点群ファイル (基準座標系)
 - 内容
 - 3D 都市モデルの建築物 LOD2、BIM モデル由来の建築物 LOD4 の統合点群データ
 - 形式
 - Pont Cloud 形式
 - データ詳細
 - ファイル出力インターフェース【IF102】を参照

- 出力
 - ✧ なし
- 機能詳細
 - 点群データ読込
 - ✧ 処理内容
 - 3D 都市モデルと BIM が統合された点群ファイル（基準座標系）を ELAS に読み込み、ドローンマップとして用いる
 - ✧ 利用するライブラリ
 - なし
 - ✧ 利用するアルゴリズム
 - なし

3. 【FN003】経路計画及び飛行指示・ウェイポイント（基準座標系）・緊急停止指示

- 機能概要
 - 3D 都市モデルを地図上に表示させた SENSYN CORE Pilot の経路計画 UI（【SC005】及び【SC006】）上でユーザーが入力した経路計画（ウェイポイント/アクション）や、モニタリング画面（【SC012】）上の緊急停止ボタンクリックによる緊急停止指示を受け取る機能
- フローチャート

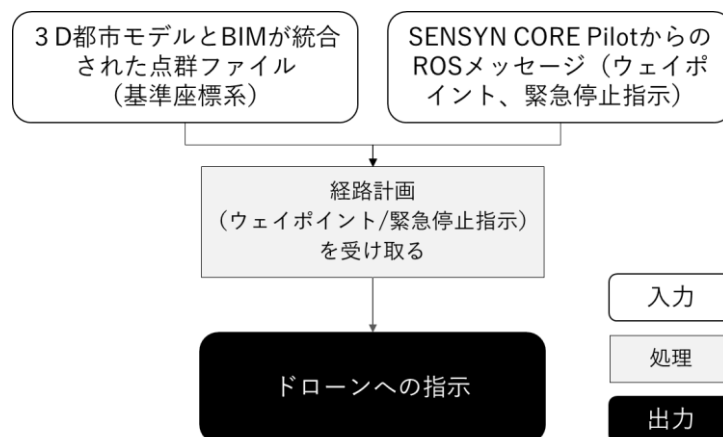


図 4-22 経路計画及び飛行指示・ウェイポイント（基準座標系）・緊急停止指示のフローチャート

- データ仕様
 - 入力
 - ✧ 3D 都市モデルと BIM が統合された点群ファイル（基準座標系）
 - 内容
 - 3D 都市モデルの建築物 LOD2、BIM モデル由来の建築物 LOD4 の統合点群データ
 - 形式
 - Point Cloud 形式
 - データ詳細
 - ファイル出力インタフェース【IF102】を参照

◇ SENSYN CORE Pilot からの ROS メッセージ（ウェイポイント、緊急停止指示）

- 内容
 - 経路計画（ウェイポイント）情報
 - 緊急停止指示
- 形式
 - ROS メッセージ形式
- データ詳細
 - 内部連携インタフェース【IF205】を参照

➢ 出力

◇ ドローンへの指示

- 内容
 - ドローンへの飛行及びアクションの指示
- 形式
 - ROS メッセージ形式
- データ詳細
 - 外部連携インタフェース【IF304】を参照

● 機能詳細

➢ 経路計画（ウェイポイント）や緊急停止指示の受け取り

◇ 処理内容

- 経路計画（ウェイポイント）や緊急停止指示を ROS メッセージで受け取る
- 受け取った指示及び点群マップを基にドローンに各種指示を送る

◇ 利用するライブラリ

- DJI Onboard SDK（ソフトウェア・ライブラリ【SL012】を参照）

◇ 利用するアルゴリズム

- なし

4. 【FN004】LiDAR ローカリゼーション

● 機能概要

- ドローンに搭載した 3D LiDAR で取得したデータと 3D 都市モデルから変換した点群マップを重ね、自己位置を推定する機能

● フローチャート

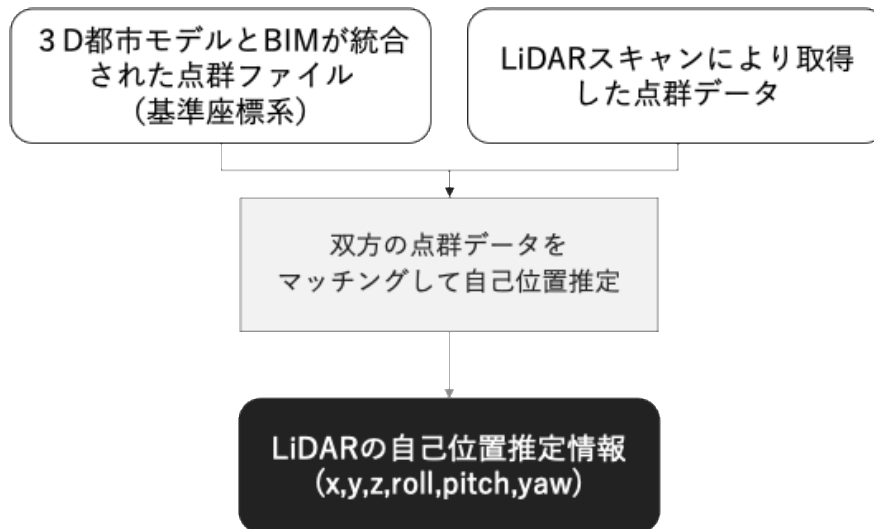


図 4-23 LiDAR ローカリゼーション機能のフローチャート

- データ仕様

- 入力

- ✧ 3D 都市モデルと BIM が統合された点群ファイル（基準座標系）

- 内容
 - 3D 都市モデルの建築物 LOD2、BIM モデル由来の建築物 LOD4 の統合点群データ
 - 形式
 - Point Cloud 形式
 - データ詳細
 - ファイル出力インターフェース【IF102】を参照

- ✧ LiDAR スキャンにより取得した点群データ

- 内容
 - LiDAR スキャンにより取得した、現実世界の点群データ
 - 形式
 - Point Cloud 形式
 - データ詳細
 - 内部連携インターフェース【IF201】を参照

- 出力

- ✧ LiDAR の自己位置推定情報(x, y, z, roll, pitch, yaw)

- 内容
 - LiDAR 機器の自己位置を推定した情報(3 次元直交座標系)
 - 形式
 - ROS メッセージ形式 ※pose 形式
 - データ詳細
 - 内部連携インターフェース【IF201】を参照
 - 生成したデータは【DT204】を参照

- 機能詳細

➤ LiDAR ローカリゼーションによる自己位置の推定

✧ 処理内容

- 3D 都市モデルと BIM が統合された点群ファイル（基準座標系）と LiDAR スキャンにより取得した点群データ双方の点群データをマッチングして自己位置推定を行う

✧ 利用するライブラリ

- Point cloud library（ソフトウェア・ライブラリ【SL006】を参照）

✧ 利用するアルゴリズム

- NDT Registration（アルゴリズム【AL001】を参照）

5. 【FN005】LiDAR 測位情報チェッカー

● 機能概要

➤ 3D LiDAR による測位情報の信頼性評価を行う機能

● フローチャート

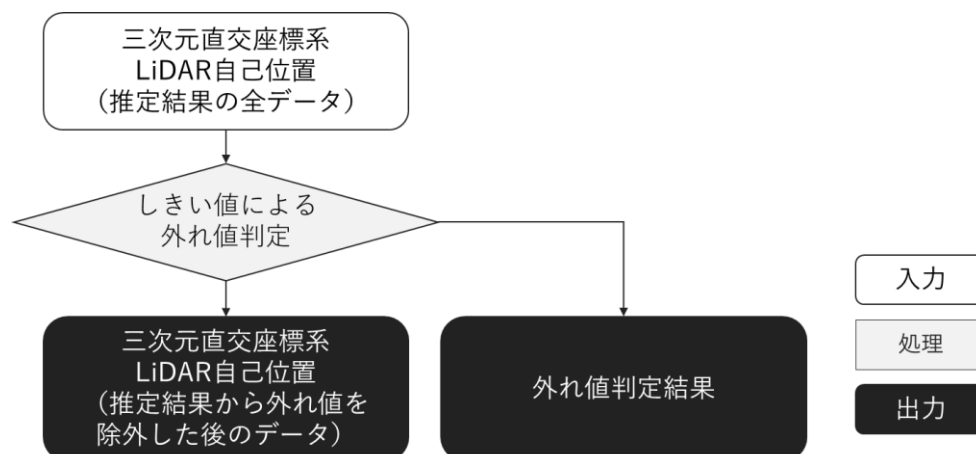


図 4-24 LiDAR 測位情報チェッカー機能のフローチャート

● データ仕様

➤ 入力

✧ 3次元直交座標系 LiDAR 自己位置（推定結果の全データ）

● 内容

- LiDAR ローカリゼーションにより出力された、LiDAR の自己位置推定情報(3次元直交座標系)

● 形式

- ROS メッセージ形式

● データ詳細

- 内部連携インターフェース【IF201】を参照

➤ 出力

✧ 3次元直交座標系 LiDAR 自己位置（推定結果から外れ値を除外した後のデータ）

● 内容

- LiDAR ローカリゼーションにより出力された、LiDAR の自己位置推定情報(3次元直

交座標系)

- 形式
 - ROS メッセージ形式
- データ詳細
 - 内部連携インタフェース【IF201】を参照

◇ 外れ値判定結果

- 内容
 - 【AL001】NDT Registration で算出した信頼性スコアが 2.0 を上回る値の場合に外れ値として設定する。
 - 信頼性スコアは、現実世界のスキャン点群と点群マップを比較し、スキャン点群の各点から最も近い点群マップ内の点までの距離の平均値を算出した値(0 以上)を示す。本判定では昨年度の実証実験を踏まえ、複数の飛行試験データを分析・検証した結果、2.0 未満であればスキャンマッチングが動作していることが確認できたため、しきい値は 2.0 とした。
- 形式
 - ROS メッセージ形式
- データ詳細
 - 内部連携インタフェース【IF201】を参照

● 機能詳細

➤ LiDAR 測位情報チェッカー

◇ 処理内容

- LiDAR ローカリゼーションの結果を測位情報統合機能に入力するかどうかの判定を、しきい値に合わせて行う

◇ 利用するライブラリ

- なし

◇ 利用するアルゴリズム

- なし

6. 【FN006】LiDAR 測位情報座標変換

● 機能概要

➤ 3D LiDAR による測位情報を基準座標系へ変換する機能

● フローチャート

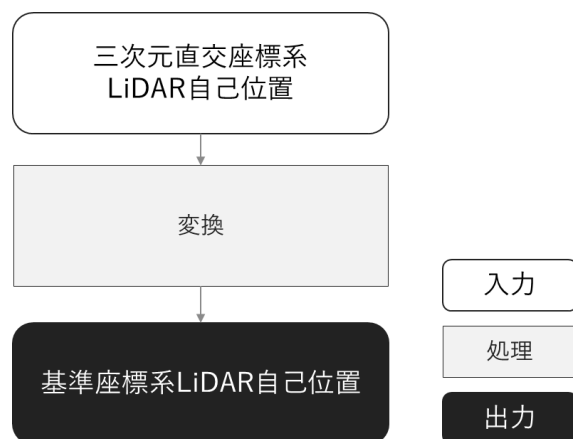


図 4-25 LiDAR 測位情報座標変換機能のフローチャート

- データ仕様
 - 入力
 - ✧ 3次元直交座標系 LiDAR 自己位置
 - 内容
 - LiDAR ローカリゼーションにより出力された、LiDAR の自己位置推定情報(3次元直交座標系)
 - 形式
 - ROS メッセージ形式
 - データ詳細
 - 内部連携インタフェース【IF201】を参照
 - 出力
 - ✧ 基準座標系 LiDAR 自己位置
 - 内容
 - LiDAR ローカリゼーションにより出力された、LiDAR の自己位置推定情報を基準座標系に直したデータ
 - 形式
 - ROS メッセージ形式
 - データ詳細
 - 内部連携インタフェース【IF206】を参照
- 機能詳細
 - 座標系の変換
 - ✧ 処理内容
 - LiDAR ローカリゼーションにより出力された自己位置情報は、ドローン用点群マップの持つ座標系での自己位置であるため、GPS 情報と統合するための基準座標系上の位置へ変換する
 - ✧ 利用するライブラリ
 - proj (ソフトウェア・ライブラリ【SL007】を参照)

◇ 利用するアルゴリズム

- なし

7. 【FN007】GNSS 測位情報チェッカー

● 機能概要

- GNSS による測位情報の信頼性評価を行う機能

● フローチャート

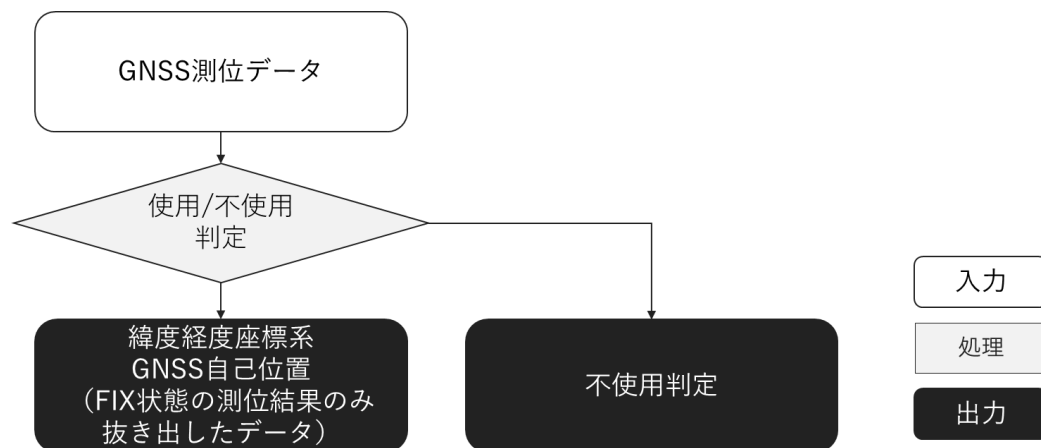


図 4-26 GNSS 測位情報チェッカー機能のフローチャート

● データ仕様

➤ 入力

◇ GNSS 測位データ

- 内容
 - GNSS 測位で得た緯度経度高さ情報
- 形式
 - NMEA 形式
- データ詳細
 - 内部連携インタフェース【IF202】を参照

➤ 出力

◇ 緯度経度座標系 GNSS 自己位置（測位結果から外れ値を除外した後のデータ）

- 内容
 - GNSS 測位で取得した自己位置（緯度経度座標系）
- 形式
 - ROS メッセージ形式
- データ詳細
 - 内部連携インタフェース【IF202】を参照

◇ 不使用判定

- 内容

- GNSS 測位で取得した自己位置（緯度経度座標系）の信頼性スコアに対して、信頼性スコアが 50 を「使用」、50 以外の値を全て「不使用」と判定するよう設定する。
- GNSS の信頼性は一般的に Fix、Float、Single で分類されており、徐々に位置精度が減少していることを示している。この値は内部的にそれぞれ 50、48、32 という値で表現されている。
- 本システムでは、GNSS の信頼性が Fix の場合のみ【FN010】測位情報統合機能に GNSS 測位結果を使用するため、GNSS 信頼性スコア 50 以外の数値を全て「不使用」と判定した。
- 形式
 - ROS メッセージ形式
- データ詳細
 - 内部連携インタフェース【IF202】を参照
- 機能詳細
 - GNSS 測位情報チェッカー
 - ✧ 処理内容
 - GNSS 測位データを測位情報統合機能に入力するかどうかの判定を、しきい値に合わせて行う
 - ✧ 利用するライブラリ
 - なし
 - ✧ 利用するアルゴリズム
 - なし

8. 【FN008】GNSS 測位情報座標変換

- 機能概要
 - GNSS による測位情報を基準座標系へ変換する機能
- フローチャート

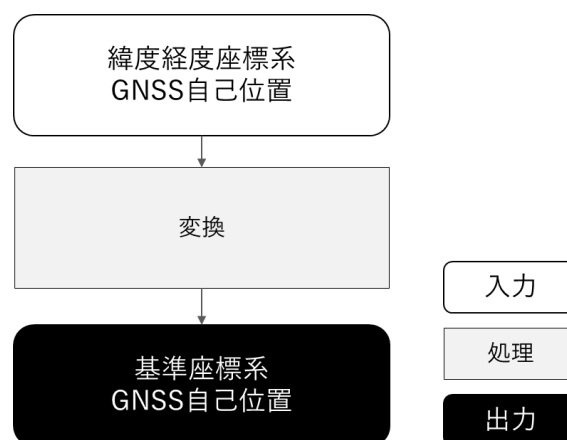


図 4-27 GNSS 測位情報座標変換機能のフローチャート

- データ仕様

➤ 入力

◇ 緯度経度座標系 GNSS 自己位置

- 内容
 - GNSS 測位で取得した自己位置（緯度経度座標系）
- 形式
 - ROS メッセージ形式
- データ詳細
 - 内部連携インタフェース【IF202】を参照

➤ 出力

◇ 基準座標系 GNSS 自己位置

- 内容
 - GNSS 測位で取得した自己位置（基準座標系）
- 形式
 - ROS メッセージ形式
- データ詳細
 - 内部連携インタフェース【IF202】を参照
 - データ詳細は【DT203】を参照

● 機能詳細

➤ 座標系の変換

◇ 処理内容

- GNSS 測位データの座標系を、緯度経度座標系から基準座標系に変換する

◇ 利用するライブラリ

- proj（ソフトウェア・ライブラリ【SL007】を参照）

◇ 利用するアルゴリズム

- なし

9. 【FN009】ビジュアルイナージナルオドメトリ（VIO）

● 機能概要

- ドローン内蔵カメラで撮影した、連続した画像の間に起こる変化、変移を捉えることで算出した移動体の移動量に加え、IMU 情報を加味して移動量を推定する機能

● フローチャート

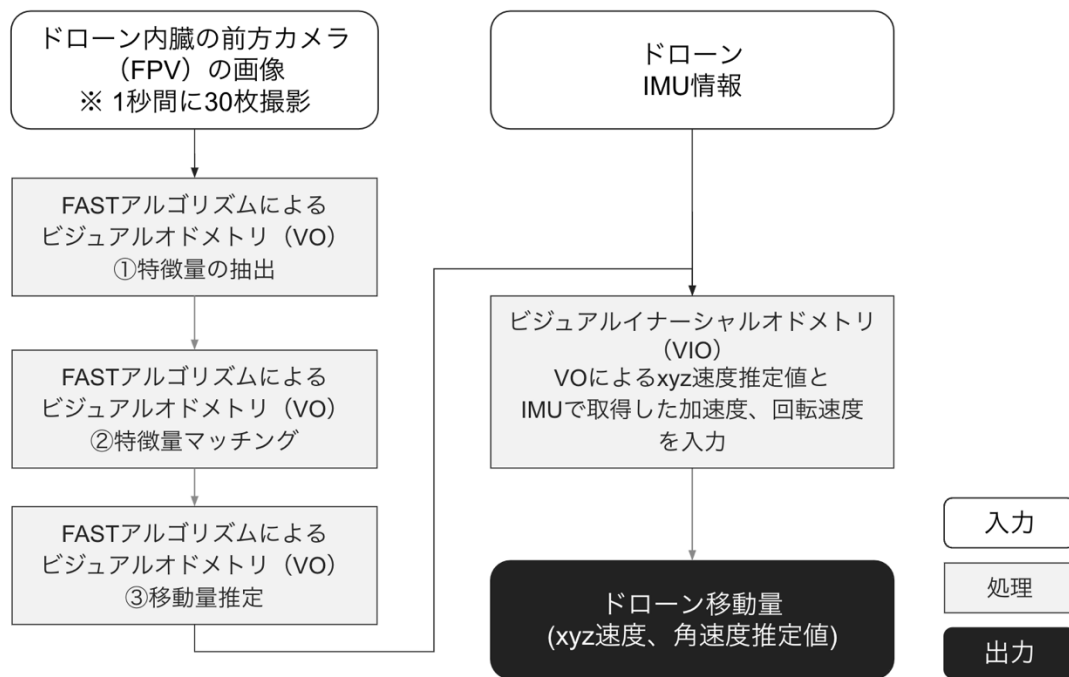


図 4-28 ビジュアルイナーシャルオドメトリ (VIO) 機能のフローチャート

- データ仕様

- 入力

- ✧ 画像データ

- 内容

- ドローン内蔵の前方カメラ (FPV) の画像 ※ 1 秒間に 30 枚撮影

- 形式

- ROS メッセージ形式 ※sensor message / image 形式

- データ詳細

- 内部連携インタフェース【IF203】を参照
 - データ詳細は【DT205】を参照

- ✧ IMU 情報

- 内容

- IMU で取得した加速度/角速度

- 形式

- ROS メッセージ形式 ※sensor message / IMU 形式

- データ詳細

- 内部連携インタフェース【IF204】を参照

- 出力

- ✧ ドローン移動量

- 内容

- 1 つ前のフレームからの移動量 (xyz 速度、角速度推定値)

- 形式

- ROS メッセージ形式 ※twist message 形式
 - データ詳細
 - 内部連携インタフェース【IF206】を参照
 - 機能詳細
 - ビジュアルオドメトリ (VO)
 - ◇ 処理内容
 - ①特徴量の抽出
カメラで撮影した画像中から特徴量を検出する
 - ②特徴量マッチング
ある時点に撮影された画像で得た特徴量と、その 1 枚後に撮影された画像から新しく得られた特徴量の対応関係を求める
 - ③移動量推定
 - “②特徴量マッチング”で特定した特徴点のペアを用い、カメラの姿勢や移動量を求める
 - ◇ 利用するライブラリ
 - OpenCV (ソフトウェア・ライブラリ【SL008】を参照)
 - ◇ 利用するアルゴリズム
 - FAST アルゴリズム (アルゴリズム【AL002】を参照)
 - ビジュアルイナージナルオドメトリ (VIO)
 - ◇ 処理内容
 - EKF(Extended Kalman Filter)にビジュアルオドメトリにより推定された速度と、IMU の加速度、回転速度を入力し、1 つ前のフレームからの移動量 (xyz 速度、角速度推定値) を出力する
 - ◇ 利用するライブラリ
 - なし
 - ◇ 利用するアルゴリズム
 - EKF (Extended Kalman Filter) (アルゴリズム【AL003】を参照)
 - ビジュアルイナージナルオドメトリ (VIO) (アルゴリズム【AL101】を参照)
10. 【FN010】測位情報統合
- 機能概要
 - 複数の測位情報を統合する機能
 - 今回の実証実験における開発では、2022 年度に開発した既存機能に【FN009】ビジュアルイナージナルオドメトリ (VIO) から出力されたデータを統合し、自己位置推定のロバスト性を向上させた
 - フローチャート

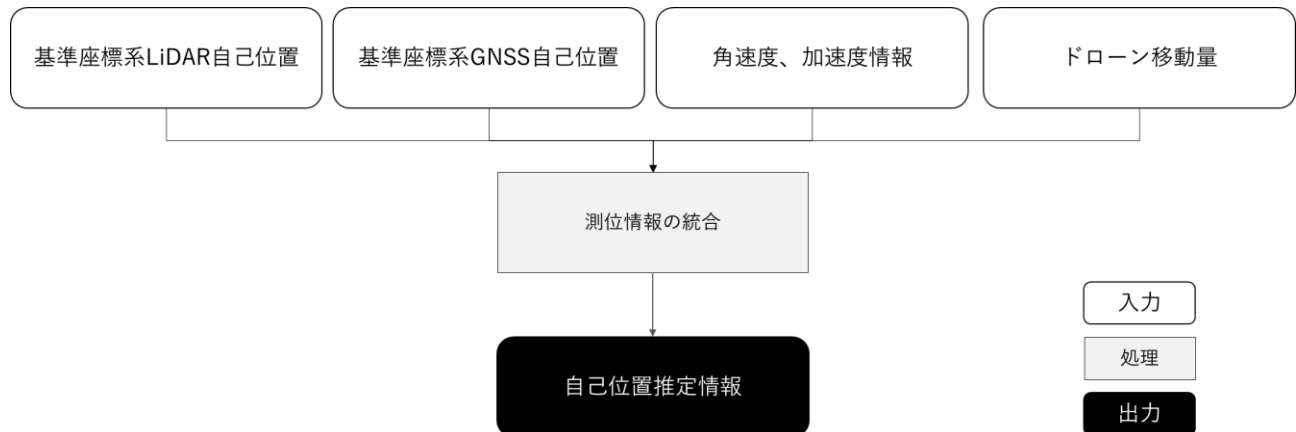


図 4-29 測位情報統合機能のフローチャート

- データ仕様

- 入力

- ✧ 基準座標系 LiDAR 自己位置

- 内容

- LiDAR ローカリゼーションにより出力された、LiDAR の自己位置推定情報を基準座標系に直したデータ

- 形式

- ROS メッセージ形式 ※pose 形式

- データ詳細

- 内部連携インタフェース【IF206】を参照

- ✧ 基準座標系 GNSS 自己位置

- 内容

- GNSS 測位で取得した自己位置（基準座標系）

- 形式

- ROS メッセージ形式 ※pose 形式

- データ詳細

- 内部連携インタフェース【IF202】を参照
- データ詳細は【DT203】を参照

- ✧ 角速度、加速度情報

- 内容

- IMU からの角速度、加速度情報

- 形式

- ROS メッセージ形式 ※IMU 形式

- データ詳細

- 内部連携インタフェース【IF204】を参照
- データ詳細は【DT204】を参照

- ✧ ドローン移動量

- 内容
 - ビジュアルイネーシャルオドメトリ (VIO) から得られる移動量 (xyz 速度、角速度推定値)
- 形式
 - ROS メッセージ形式 ※twist message 形式
- データ詳細
 - 内部連携インタフェース【IF206】を参照
 - 開発したアルゴリズム【AL101】を参照
- 出力
 - ◇ 自己位置推定情報
 - 内容
 - 各種測位情報が統合された結果、推定される自己位置情報
 - 形式
 - ROS メッセージ形式 ※odometry 形式、pose 形式
 - データ詳細
 - 開発したアルゴリズム【AL102】を参照
- 機能詳細
 - 測位情報の統合
 - ◇ 処理内容
 - LiDAR ユニット、GNSS ユニット、IMU ユニット、カメラユニット (ビジュアルイネーシャルオドメトリ (VIO)) から得られた複数の測位情報を統合する
 - ◇ 利用するライブラリ
 - なし
 - ◇ 利用するアルゴリズム
 - EKF (Extended Kalman Filter) (アルゴリズム【AL003】を参照)
 - 測位情報の統合アルゴリズム (アルゴリズム【AL102】を参照)

11. 【FN011】障害物回避

- 機能概要
 - 障害物が存在する場合は障害物を避ける経路を生成し、移動目標点を更新する機能
- フローチャート

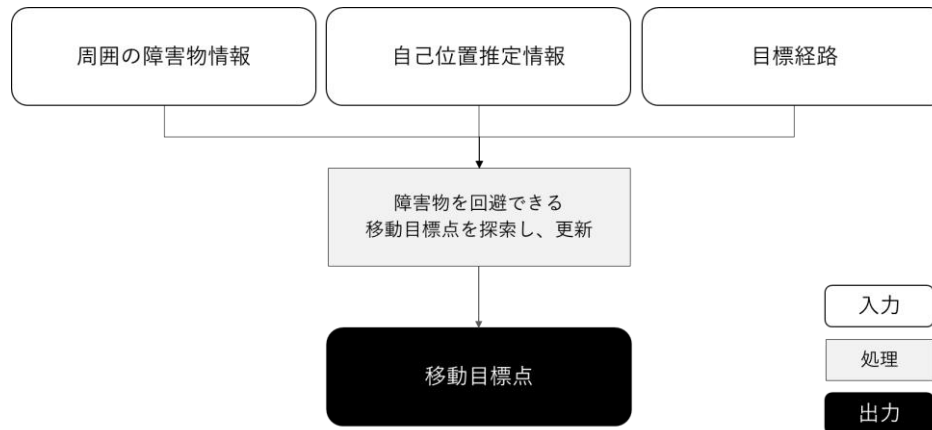


図 4-30 障害物回避機能のフローチャート

- データ仕様
 - 入力
 - ◇ 周囲の障害物情報
 - 内容
 - 機体に搭載した 3D LiDAR から得られるスキャン結果を基に得られた周囲の障害物情報
 - 形式
 - ROS メッセージ形式
 - データ詳細
 - 内部連携インタフェース【IF201】を参照
 - データ詳細は【DT202】を参照
 - ◇ 自己位置推定情報
 - 内容
 - 測位情報統合の結果得られた自己位置推定情報
 - 形式
 - ROS メッセージ形式 ※odometry 形式、pose 形式
 - データ詳細
 - 開発したアルゴリズム【AL102】を参照
 - ◇ 目標経路
 - 内容
 - ドローンが飛行する目標経路
 - 形式

- ROS メッセージ形式
 - データ詳細
 - 内部連携インタフェース【IF205】を参照
- 出力
 - ◇ 移動目標点
 - 内容
 - ドローンの自己位置と目標経路から決めた移動目標点を、障害物を回避するように更新する
 - 形式
 - ROS メッセージ形式
 - データ詳細
 - 内部連携インタフェース【IF205】を参照
- 機能詳細
 - 障害物判定
 - ◇ 処理内容
 - 機体に搭載した 3D LiDAR から得られるスキャン結果を基に、30cm 四方に区切った空間内（ボクセル）に点群が存在するか判定を行う
 - 点群を検知した場合は障害物がある空間として扱う
 - ◇ 利用するライブラリ
 - Octomap（ソフトウェア・ライブラリ【SL009】を参照）
 - ompl（ソフトウェア・ライブラリ【SL010】を参照）
 - ◇ 利用するアルゴリズム
 - なし
 - 障害物回避
 - ◇ 処理内容
 - 障害物が存在する場合は障害物を避ける経路を生成し、移動目標点を更新する
 - ◇ 利用するライブラリ
 - Octomap（ソフトウェア・ライブラリ【SL009】を参照）
 - ompl（ソフトウェア・ライブラリ【SL010】を参照）
 - ◇ 利用するアルゴリズム
 - なし

12. 【FN012】目標速度算出

- 機能概要
 - 自己位置と目標位置からドローンの移動速度を決定/指示する機能
- フローチャート

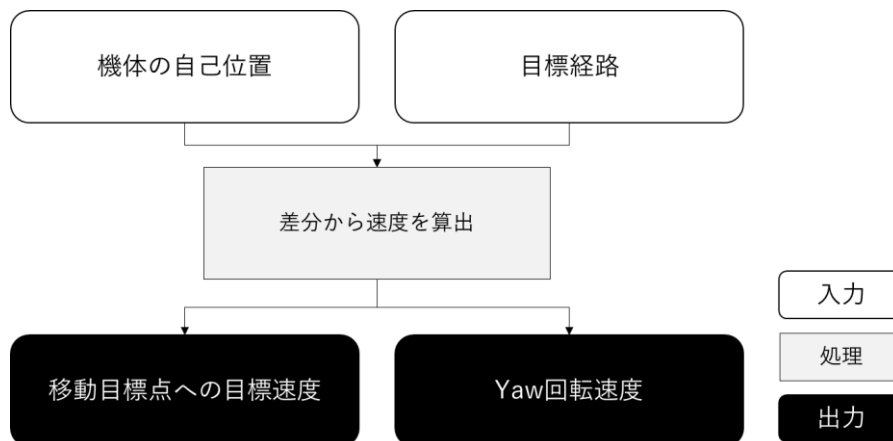


図 4-31 目標速度算出機能のフローチャート

- データ仕様
 - 入力
 - ◇ 機体の自己位置
 - 内容
 - 測位情報統合の結果から得られた自己位置推定情報
 - 形式
 - ROS メッセージ形式 ※odometry 形式、pose 形式
 - データ詳細
 - 開発したアルゴリズム【AL102】を参照
 - ◇ 目標経路
 - 内容
 - ドローンが飛行する目標経路
 - 形式
 - ROS メッセージ形式
 - データ詳細
 - 内部連携インタフェース【IF205】を参照
 - 出力
 - ◇ 移動目標点への目標速度
 - 内容
 - ドローンの自己位置と移動目標点の差分から定まった、目標方向に進む進行方向への速度
 - 形式
 - ROS メッセージ形式

- データ詳細
 - 内部連携インタフェース【IF205】を参照
- ✧ Yaw 回転速度
 - 内容
 - ドローンの自己位置と移動目標点の差分から定まった、目標方向に進む角速度
 - 形式
 - ROS メッセージ形式
 - データ詳細
 - 内部連携インタフェース【IF204】を参照
- 機能詳細
 - 目標速度の算出
 - ✧ 処理内容
 - ドローンの自己位置と目標経路から移動目標点を求め、目標速度を算出する
 - ✧ 利用するライブラリ
 - なし
 - ✧ 利用するアルゴリズム
 - なし

13. 【FN013】機体制御

- 機能概要
 - 飛行指示に従い、機体を制御する機能
- フローチャート

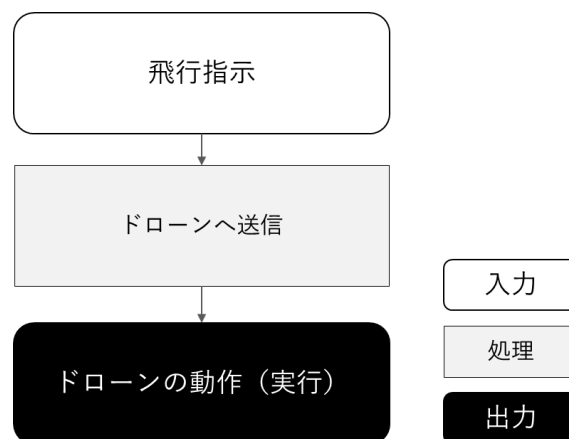


図 4-32 機体制御のフローチャート

- データ仕様
 - 入力
 - ✧ 飛行指示
 - 内容
 - ELAS からのドローンに送られる飛行指示

- 形式
 - DJI Onboard SDK メッセージ形式
- データ詳細
 - 内部連携インタフェース【IF205】を参照
- 出力
 - ◇ ドローンの動作（実行）
 - 内容
 - 現実世界でのドローンの動作（自律飛行、緊急停止）
 - 形式
 - なし
 - データ詳細
 - なし
- 機能詳細
 - 機体の制御
 - ◇ 処理内容
 - 飛行指示に従い機体を制御する
 - ◇ 利用するライブラリ
 - DJI Onboard SDK（ソフトウェア・ライブラリ【SL012】を参照）
 - ◇ 利用するアルゴリズム
 - なし

14. 【FN014】 地形データ切り替え

● 機能概要

- 3D 都市モデルと BIM の統合マップを SENSYN CORE Pilot 画面の 3D マップ（Cesium）上に表示するために、3D 都市モデルが存在する場所では利用する地形データを Cesium のデフォルトの地形データから PLATEAU-Terrain が持つ地形データへ切り替えを行う機能

● フローチャート

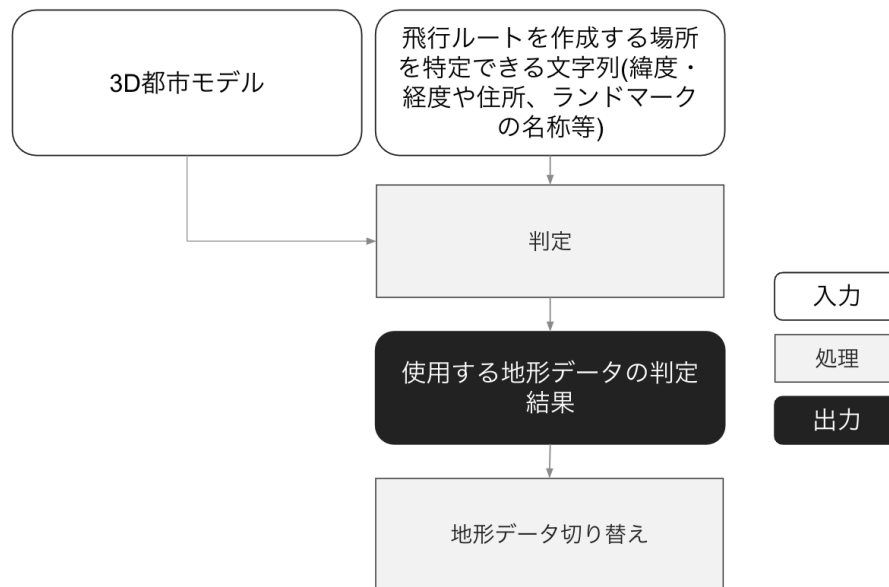


図 4-33 地形データ切り替え機能のフローチャート

● データ仕様

➤ 入力

- ✧ 飛行ルートを作成する場所を特定できる文字列(緯度・経度や住所、ランドマークの名称等)

● 内容

- 緯度・経度の組や住所、ランドマークの名称等、飛行ルートを作成する場所を示す情報
- 住所や建物名はジオコーディングし、最終的には緯度・経度の組として扱う

● 形式

- 文字列形式

● データ詳細

- 外部連携インターフェース【IF301】【IF302】を参照
- 場所を特定できる文字列を緯度・経度座標に変換するためのジオコードデータベースを利用（外部連携インターフェース【IF303】を参照）
- 利用したその他データは【DT102】【DT103】を参照

➤ 出力

- ✧ 使用する地形データの判定結果

● 内容

- Cesium のデフォルト地形データと PLATEAU-Terrain のどちらを使用するかを意味

する列挙値

- 形式
 - 文字列形式又は数値形式
- データ詳細
 - なし
- 機能詳細
 - 地形データ切り替え
 - ✧ 処理内容
 - 3D 都市モデルと BIM の統合マップを SENSYN CORE Pilot 上で表示するために地形データの切り替えを行う
 - Cesium のデフォルトの地形データ上に 3D 都市モデルを表示させると、対地高度の前提となる標高データが異なり、3D 都市モデルが地面に埋まる又は地表から浮いた状態で表示される場合があるため、この問題を解決するために Cesium 上で利用する地形データを適切に置き換える
 - SENSYN CORE Pilot 上で PLATEAU-Terrain を使用せずに作成したルートに影響を与えないようにする
 - ただし、PLATEAU-Terrain が持つ地形情報は日本国内のみであるため、日本国外で飛行経路を作成する場合は Cesium のデフォルト地形データを利用する必要がある
 - この切り替えを自動で行う
 - ✧ 利用するライブラリ
 - Cesium Geocode API（ソフトウェア・ライブラリ【SL013】を参照）
 - ✧ 利用するアルゴリズム
 - 日本国内を囲む緯度・経度座標のリストを利用し、指定の緯度・経度座標が囲まれた中に存在するかを判定するアルゴリズム（アルゴリズム【AL103】を参照）
 - PLATEAU-Terrain を用いるルートかどうかを判定するアルゴリズム（アルゴリズム【AL104】を参照）

15. 【FN015】経路計画 UI

- 機能概要
 - 車両・ドローンの経路設定及び設定した経路を MAVLink 形式に変換する機能
- フローチャート

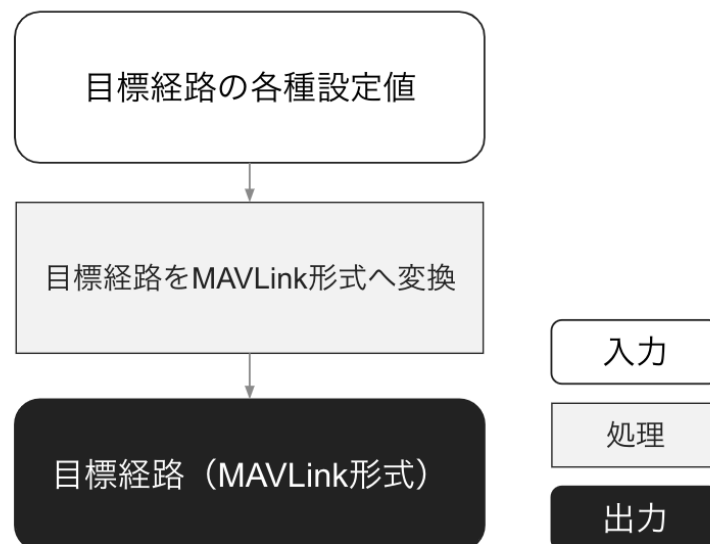


図 4-34 経路計画 UI のフローチャート

- データ仕様
 - 入力
 - ◇ 目標経路の各種設定値
 - 内容
 - 目標経路（飛行ルート）の情報（各ウェイポイントの緯度経度高さ、各ウェイポイント間の移動速度）
 - 形式
 - 数値形式
 - データ詳細
 - 内部連携インタフェース【IF205】を参照
 - 出力
 - ◇ 目標経路（MAVLink 形式）
 - 内容
 - 目標経路（飛行ルート）のルート情報
 - 形式
 - MAVLink 形式 ※WayPoints
 - データ詳細
 - 内部連携インタフェース【IF205】を参照
- 機能詳細
 - 経路計画 UI

- ✧ 処理内容
 - ユーザーが UI 上で入力した飛行ルート各情報を MAVLink 形式に変換する
- ✧ 利用するライブラリ
 - MAVLink toolchain (ソフトウェア・ライブラリ【SL015】を参照)
- ✧ 利用するアルゴリズム
 - なし

16. 【FN016】 PLATEAU-Terrain 対応機能

- 機能概要
 - SENSYN CORE Pilot を用いてドローンの経路計画を行う際に、SENSYN CORE Pilot で用いられている標高データと異なる標高データ (PLATEAU-Terrain) を用いた場合、ドローンへのルート転送時に標高差を補正する機能
- フローチャート

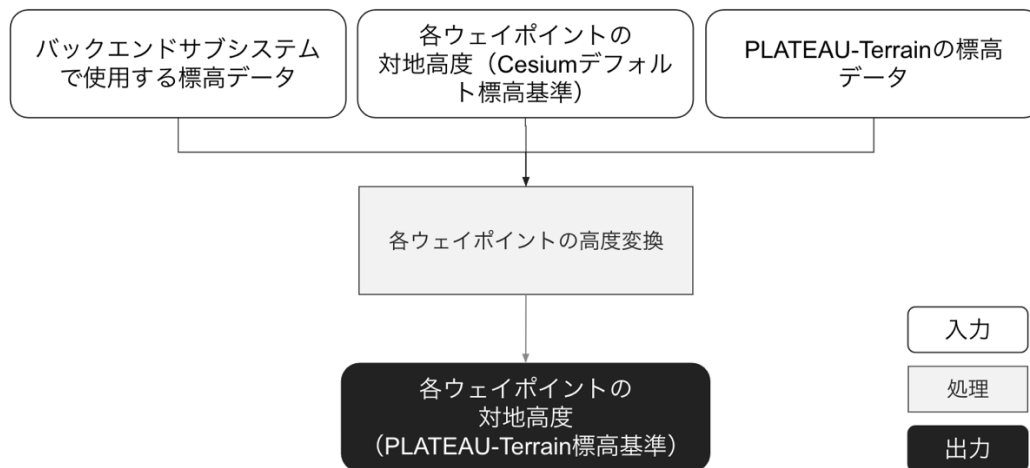


図 4-35 PLATEAU-Terrain 対応機能のフローチャート

- データ仕様
 - 入力
 - ✧ 各ウェイポイントの対地高度 (Cesium デフォルト標高基準)
 - 内容
 - Cesium の標高データを基準とする、各ウェイポイントの対地高度
 - 形式
 - 数値形式
 - データ詳細
 - 外部連携インタフェース【IF301】を参照
 - ✧ PLATEAU-Terrain の標高データ
 - 内容
 - PLATEAU-Terrain が持つ標高データ
 - 形式

- PLATEAU-Terrain 形式
- データ詳細
 - 外部連携インタフェース【IF302】を参照
- ✧ バックエンドサブシステムが使用する標高データ
 - 内容
 - バックエンドサブシステムが持つ標高データ
 - 形式
 - quantized-mesh 形式
 - データ詳細
 - バックエンドサブシステムで使用する標高データ ※SENSYN CORE Pilot ベースで開発を行うため、詳細は開示不可
- 出力
 - ✧ 各ウェイポイントの対地高度（バックエンドサブシステムの標高基準）
 - 内容
 - バックエンドサブシステムが利用している標高データを基準とする、飛行経路上の各ウェイポイントの対地高度
 - 形式
 - 数値形式
 - データ詳細
 - 外部連携インタフェース【IF301】を参照
- 機能詳細
 - 各ウェイポイントの高度変換
 - ✧ 処理内容
 - 3D 都市モデルと BIM の統合マップ（PLATEAU-Terrain に合わせたデータ）を基に SENSYN CORE Pilot を用いてドローンの経路計画を行う際に、バックエンドサブシステム上で用いられている標高データと PLATEAU-Terrain の標高データを整合させる
 - SENSYN CORE Pilot において経路計画（ルート作成）時に利用するバックエンドサブシステムの標高データ（TP 高度ではない、SRTM 等の標高データ）は PLATEAU-Terrain のデータ（TP 高度）とは異なる
 - 利用する標高データ間の差分はドローンの飛行に影響を与えるため、各ウェイポイントの高度を適切に変換し、異なる標高データを用いても意図どおりの高度を保つことができるよう処理を行う
 - ✧ 利用するライブラリ
 - PLATEAU-Terrain の標高データ（外部連携インタフェース【IF302】を参照）
 - ✧ 利用するアルゴリズム
 - なし

17. 【FN017】 ROS メッセージ双方向変換

- 機能概要

- SENSYN CORE Pilot と ROS により動作するロボットの間で、情報を双方向にやり取りするのに適切な形式に変換する機能

- フローチャート

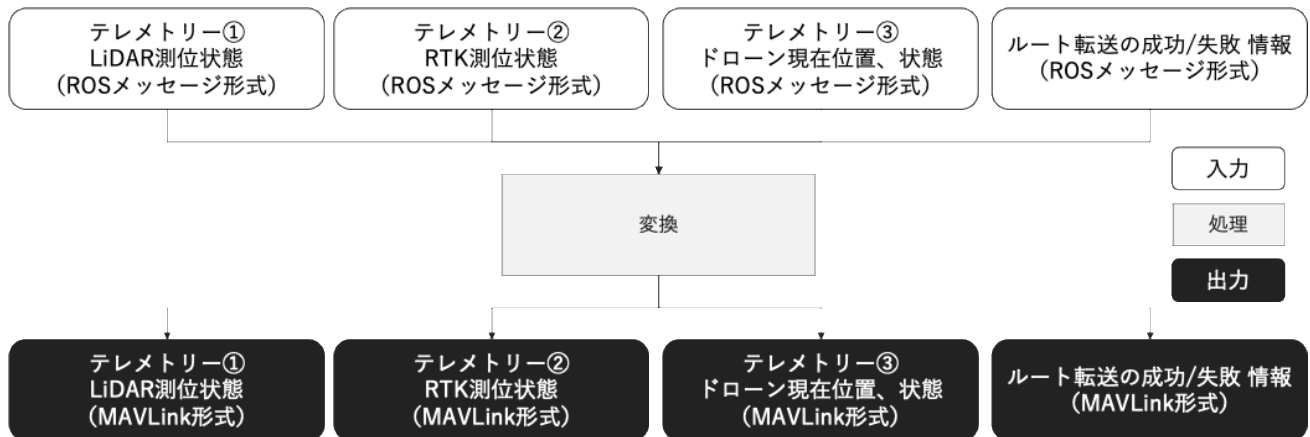


図 4-36 ROS メッセージ双方向変換機能のフローチャート（ドローンから SENSYN CORE Pilot への入力）

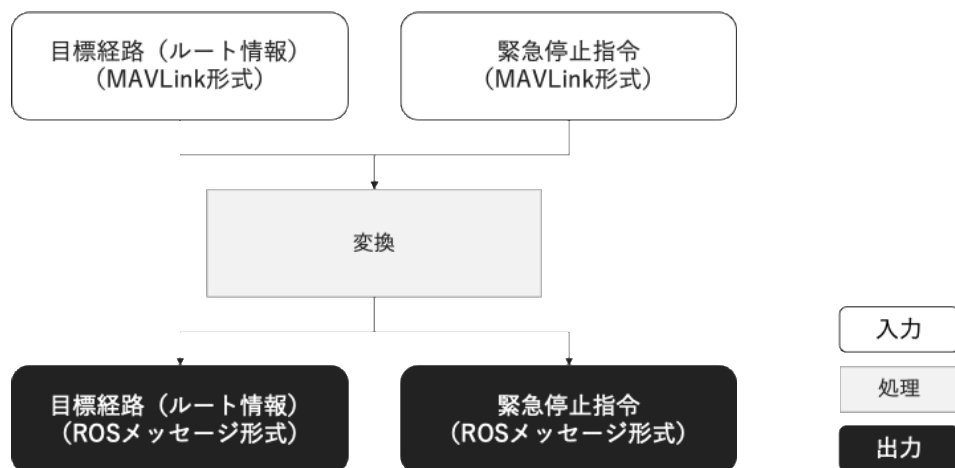


図 4-37 ROS メッセージ双方向変換機能のフローチャート（SENSYN CORE Pilot からドローンへの入力）

- データ仕様

- 入力

- ✧ ドローン（ELAS）からの入力

- 内容

- テレメトリー① LiDAR 測位状態
- テレメトリー② RTK 測位状態
- テレメトリー③ ドローン現在位置、状態
- ルート転送の成功/失敗 情報

- 形式

- ROS メッセージ形式
 - データ詳細
 - 内部連携インタフェース【IF205】を参照
- ✧ SENSYN CORE Pilot からの入力
 - 内容
 - 目標経路（ルート情報）
 - 緊急停止指令
 - 形式
 - MAVLink 形式
 - データ詳細
 - 内部連携インタフェース【IF205】を参照
- 出力
 - ✧ ドローン（ELAS）への出力
 - 内容
 - 目標経路（ルート情報）
 - 緊急停止指令
 - 形式
 - ROS メッセージ形式
 - データ詳細
 - 内部連携インタフェース【IF205】を参照
 - ✧ SENSYN CORE Pilot への出力
 - 内容
 - テレメトリー①LiDAR 測位状態
 - テレメトリー②RTK 測位状態
 - テレメトリー③ドローン現在位置、状態
 - ルート転送の成功/失敗 情報
 - 形式
 - MAVLink 形式
 - データ詳細
 - 内部連携インタフェース【IF205】を参照
- 機能詳細
 - ROS メッセージ双方向変換
 - ✧ 処理内容
 - SENSYN CORE Pilot と ROS により動作するロボットの間で情報を双方向にやり取りするため、送信するメッセージのメッセージ形式を送信先に合わせて ROS メッセージ形式、MAVLink 形式のそれぞれ適切な形式に変換する
 - ✧ 利用するライブラリ
 - Boost（ソフトウェア・ライブラリ【SL011】を参照）

✧ 利用するアルゴリズム

- テレメトリーについては、辞書ベースのデータマッピングアルゴリズム（アルゴリズム【AL004】を参照）
- ルート転送については、State machine ベースの変換アルゴリズム（アルゴリズム【AL005】を参照）

18. 【FN018】MQTT メッセージング双方向変換

- 機能概要
 - SENSYN CORE Pilot と車両の間で、情報を双方向にやり取りするのに適切な形式に変換する機能
- フローチャート

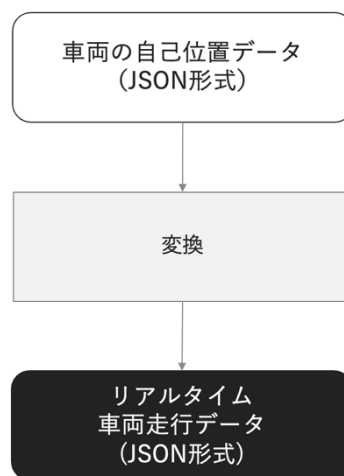


図 4-38 MQTT メッセージング双方向変換機能のフローチャート（車両から SENSYN CORE Pilot）

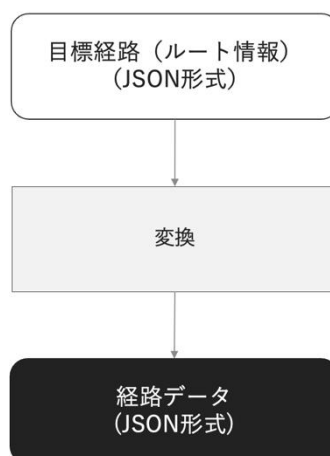


図 4-39 MQTT メッセージング双方向変換機能のフローチャート（SENSYN CORE Pilot から車両）

- データ仕様
 - 入力
 - ◇ 車両からの入力
 - 内容
 - 車両の自己位置データ
 - 形式
 - JSON 形式
 - データ詳細
 - 内部連携インタフェース【IF205】を参照
 - ◇ SENSYN CORE Pilot からの入力
 - 内容
 - 目標経路（ルート情報）
 - 形式
 - JSON 形式
 - データ詳細
 - 内部連携インタフェース【IF205】を参照
 - 出力
 - ◇ SENSYN CORE Pilot への出力
 - 内容
 - リアルタイム車両走行データ
 - 形式
 - JSON 形式
 - データ詳細
 - 内部連携インタフェース【IF205】を参照
 - ◇ 車両への出力
 - 内容
 - 経路データ
 - 形式
 - JSON 形式
 - データ詳細
 - 内部連携インタフェース【IF205】を参照
- 機能詳細
 - MQTT メッセージング双方向変換
 - ◇ 処理内容
 - SENSYN CORE Pilot と車両の間で、情報を双方向にやり取りするため、メッセージ形式

を送信先に合わせて JSON 形式のそれぞれ適切な形式に変更する

- ✧ 使用するライブラリ
 - なし
- ✧ 使用するアルゴリズム
 - なし

19. 【FN019】車両・ドローン位置可視化

- 機能概要
 - 車両・ドローンの現在位置をリアルタイムで表示する機能
- フローチャート

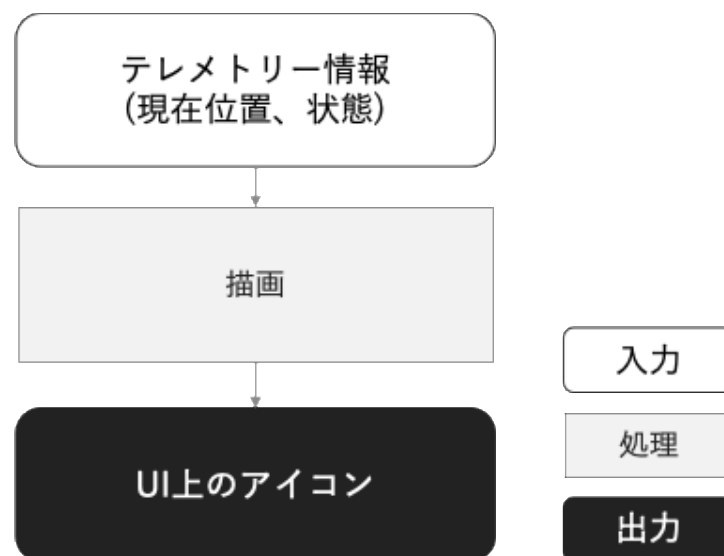


図 4-40 車両・ドローン位置可視化機能のフローチャート

- データ仕様
 - 入力
 - ✧ テレメトリ情報
 - 内容
 - 車両・ドローンの現在位置、状態を示すテレメトリ情報
 - 形式
 - MAVLink 形式
 - データ詳細
 - 内部連携インタフェース【IF205】を参照
 - 出力
 - ✧ SENSYN CORE Pilot の地図上でのドローンアイコン描画
 - 内容
 - 車両・ドローン位置を示すアイコンにより、車両・ドローンの現在位置を Cesium 上で描画する

- 形式
 - GLB 形式
- データ詳細
 - 内部連携インタフェース【IF205】を参照
- 機能詳細
 - ドローン位置の可視化
 - ◇ 処理内容
 - 車両・ドローン（ELAS）から受信したテレメトリ情報を、UI 上に描画する
 - ◇ 利用するライブラリ
 - PLATEAU-Terrain の標高データ（外部連携インタフェース【IF302】を参照）
 - egm96-universal（ソフトウェア・ライブラリ【SL014】を参照）
 - ◇ 利用するアルゴリズム
 - なし

20. 【FN020】緊急停止

- 機能概要
 - ドローンを緊急停止させる指令を ELAS に送る機能
- フローチャート

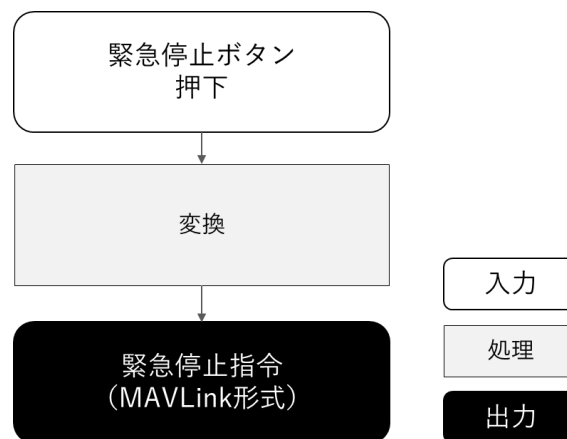


図 4-41 緊急停止機能のフローチャート

- データ仕様
 - 入力
 - ◇ ユーザーによる UI からの入力（ボタンクリック）
 - 内容
 - UI 上のドローン緊急停止ボタンをクリック
 - 形式
 - 数値形式
 - データ詳細
 - 内部連携インタフェース【IF205】を参照

- 出力
 - ✧ 緊急停止指令（MAVLink 形式）
 - 内容
 - ドローンに緊急停止を指示する命令
 - 形式
 - MAVLink 形式
 - データ詳細
 - 内部連携インタフェース【IF205】を参照

- 機能詳細

- 緊急停止
 - ✧ 処理内容
 - UI 上でのボタンクリックを MAVLink 形式の命令に変換する
 - ✧ 利用するライブラリ
 - SENSYN CORE Pilot ベースで開発を行うため、詳細は開示不可。
 - ✧ 利用するアルゴリズム
 - なし

2) 車両用機能一覧

21. 【FN101】点群生成

- 機能概要
 - 3D 都市モデルから点群マップを生成する機能
- フローチャート

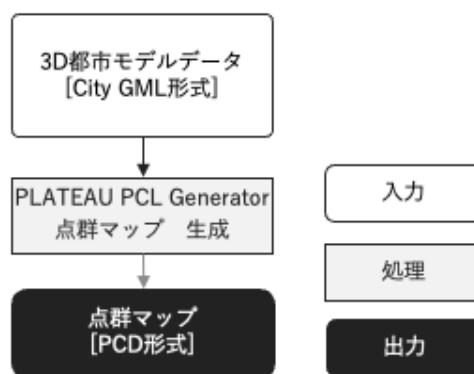


図 4-42 点群生成のフローチャート

- データ仕様
 - 入力
 - ✧ 3D 都市モデルデータ
 - 内容
 - 対象エリアの 3D 都市モデルを入力

- 形式
 - CityGML 形式
- データ詳細
 - 利用した 3D 都市モデル【DT001 ～ DT005】を参照
- ◇ PCL Generator のパラメーター
 - 内容
 - 点群生成用の車両、LiDAR パラメーターを入力
 - 形式
 - なしデータ詳細
 - 内部連携インタフェース【IF251】を参照
- 出力
 - ◇ 点群マップ
 - 内容
 - PCL Generator から出力された点群マップ
 - 形式
 - PCD 形式
 - データ詳細
 - ファイル入力インタフェース【IF102】を参照
- 機能詳細
 - 点群マップ生成
 - ◇ 処理内容
 - 仮想空間内に配置された 3D 都市モデル内を走行して、点群マップを生成する機能
 - ◇ 利用するライブラリ
 - PLATEAU SDK for Unity
 - PLATEAU Point Cloud Generator
 - ◇ 利用するアルゴリズム
 - なし
 - ◇ 生成したデータは【DT201】を参照
 - ◇ ドローン側の 3D 都市モデルの点群転換【FN001】との違い

車両側の自動運転では、現実の車両が走行するルートに近い点群マップを生成するために、仮想空間上に 3D 都市モデル、車両を読み込み、仮想空間上を走行することで点群マップの生成を行う

※ドローンの自律飛行では、【FN001】により CityGML 形式の 3D 都市モデルを直接点群化し、点群マップとして用いる。そのため本機能は用いない。

22. 【FN102】点群データ読み込み

- 機能概要

➤ LiDAR の点群データを Autoware に取り込む機能

● フローチャート

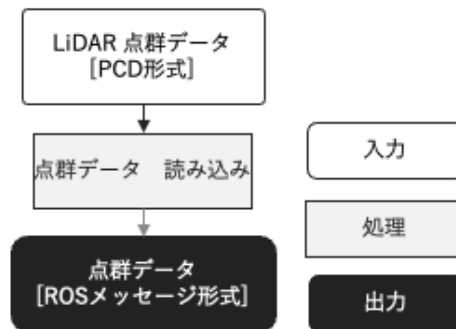


図 4-43 点群データ読み込みのフローチャート

➤ 入力

✧ LiDAR 点群データ

- 内容
 - 3D LiDAR から得られる点群データ
- 形式
 - PCD 形式
- データ詳細
 - 内部連携インタフェース【IF261】を参照

➤ 出力

✧ 点群データ

- 内容
 - LiDAR から得られたセンサーデータを ROS メッセージ形式に変換し、Autoware の自己位置推定処理で使用する
- 形式
 - ROS メッセージ
- データ詳細
 - 内部連携インタフェース【IF261】を参照

● 機能詳細

➤ LiDAR の点群データを取り込む機能

✧ 処理内容

- 3D LiDAR から取得した PCD 形式の点群データを ROS メッセージに変換する機能

✧ 利用するライブラリ

- ROS2-ouster（ソフトウェア・ライブラリ【SL108 を参照】）

✧ 利用するアルゴリズム

- なし

23. 【FN103】NMEA データ読込

- 機能概要
 - RTK 測位デバイスから NMEA 形式のデータを取り込む機能
- フローチャート

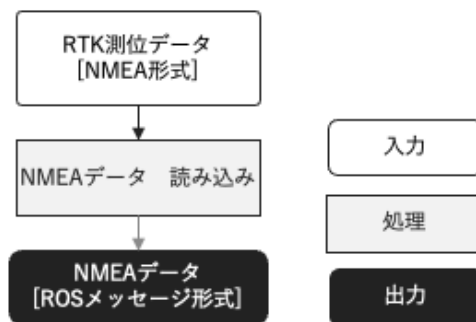


図 4-44 NMEA データ読み込みのフローチャート

- データ仕様
 - 入力
 - ✧ NMEA データ
 - 内容
 - RTK 測位デバイスから得られるデータ
 - 形式
 - NMEA 形式
 - データ詳細
 - 内部連携インタフェース【IF262】を参照
 - 出力
 - ✧ NMEA データ
 - 内容
 - RTK 測位デバイスから得られた NMEA 形式のデータを Autoware の自己位置推定処理で使用するために ROS メッセージに変換
 - 形式
 - ROS メッセージ
 - データ詳細
 - 内部連携インタフェース【IF262】を参照
- 機能詳細
 - NMEA データ読込
 - ✧ 処理内容
 - RTK 測位デバイスから NMEA 形式のデータを受け取る
 - 取得した NMEA 形式のデータを ROS メッセージに変換
 - ✧ 利用するライブラリ
 - nmea_navsat_driver (ソフトウェア・ライブラリ【SL110】を参照)
 - ✧ 利用するアルゴリズム

- なし

24. 【FN104】 センサーデータ処理

- 機能概要
 - 車両に搭載した LiDAR センサーからセンサーデータの取得と処理を行う機能
- フローチャート

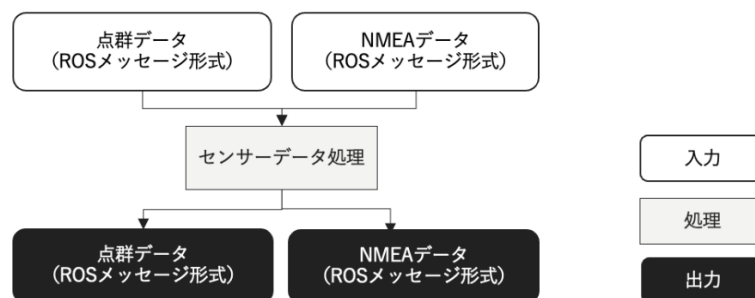


図 4-45 センサーデータ処理のフローチャート

- データ仕様
 - 入力
 - ◇ LiDAR 点群データ
 - 内容
 - ROS2-ouster から得られる、3D LiDAR から得られる点群データを ROS メッセージに変換したデータ
 - 形式
 - ROS メッセージ
 - データ詳細
 - 内部連携インタフェース【IF263】を参照
 - ◇ NMEA データ
 - 内容
 - RTK 測位デバイスから得られた NMEA 形式のデータを LiDAR 点群データと併用し、自己位置推定処理を行う Autoware に読み込ませるために、ROS メッセージに変換
 - 形式
 - ROS メッセージ
 - データ詳細
 - 内部連携インタフェース【IF262】を参照
 - 出力
 - ◇ LiDAR 点群データ
 - 内容
 - 点群間引き処理やノイズ除去を行った後の 3D LiDAR のデータ

- 形式
 - ROS メッセージ
- データ詳細
 - 内部連携インタフェース【IF263】を参照
- ✧ NMEA データ
 - 内容
 - 衛星数などのメタデータを除いた自己位置測位に必要な NMEA 形式のデータ
 - 形式
 - ROS メッセージ
 - データ詳細
 - 内部連携インタフェース【IF262】を参照
- 機能詳細
 - センサーデータ処理
 - ✧ 処理内容
 - LiDAR から取得したデータを ROS メッセージで受け取る
 - 車体に当たる点群の領域を選択し、点群データのマスク処理を行う
 - 点群のノイズ除去を行う
 - ✧ 利用するライブラリ
 - Autoware-sensing（ソフトウェア・ライブラリ【SL112】を参照）
 - ✧ 利用するアルゴリズム
 - なし

25. 【FN105】自己位置測位

- 機能概要
 - 3D LiDAR の点群データと点群マップのマッチングを行い、自己位置測位を行う機能
- フローチャート

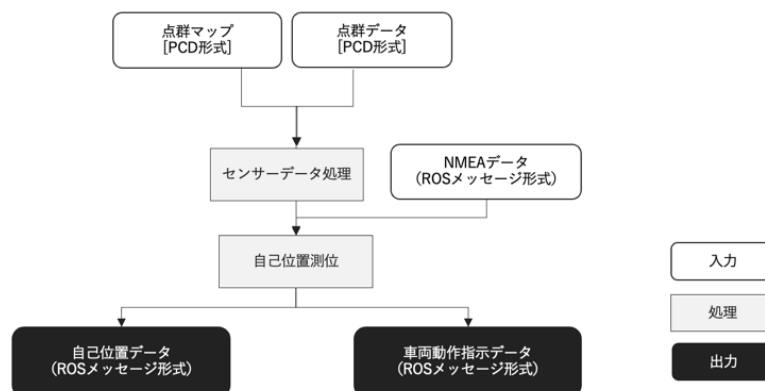


図 4-46 自己位置測位のフローチャート

- データ仕様

- 入力

- ✧ 点群データ

- 内容
 - Autoware-sensing から出力された点群データの ROS メッセージ
- 形式
 - ROS メッセージ
- データ詳細
 - 内部連携インタフェース【IF261】を参照

- ✧ NMEA データ（ROS メッセージ形式）

- 内容
 - RTK 測位デバイスから得られた NMEA 形式のデータを Autoware の自己位置推定処理で使用するために ROS メッセージに変換
- 形式
 - ROS メッセージ
- データ詳細
 - 内部連携インタフェース【IF262】を参照

- ✧ 点群マップ

- 内容
 - PLATEAU PCL Generator から出力された点群マップ
- 形式
 - PCD 形式
- データ詳細
 - ファイル入力インタフェース【IF102】を参照

- 出力

- ✧ 自己位置精度データ（ROS メッセージ形式）

- 内容
 - 自己位置測位の精度割合を表したデータ
- 形式
 - ROS メッセージ
- データ詳細
 - 内部連携インタフェース【IF264】を参照
 - 精度の判断に関しては、50%以上の精度を高、25%から 50%の精度を中、25%以下の精度を低とする。

- ✧ 車両動作指示データ（ROS メッセージ形式）

- 内容
 - 車両の動作指示を表したデータ

- 形式
 - ROS メッセージ
- データ詳細
 - 内部連携インタフェース【IF265】を参照
- 機能詳細
 - 自己位置測位
 - ◇ 昨年度からの改善点
 - 昨年度は点群マッチングのみを利用とする自己位置測位だったが、今年度は点群マッチングに加え、RTK 測位データも加えることで精度の向上を狙う
 - ◇ 処理内容
 - 点群マップと、LiDAR で取得した点群データとマッチングを行う
 - RTK 測位データを用いて、点群マッチングの測位情報を補完する
 - ◇ 利用するライブラリ
 - Autoware-localization（ソフトウェア・ライブラリ【SL113】を参照）
 - ◇ 利用するアルゴリズム
 - NDT Registration（アルゴリズム【AL001】を参照）

26. 【FN106】自動走行処理

- 機能概要
 - 車両の動作指示を受け取り、車両への動作命令（走行、ステアリング、ブレーキ処理）を出力する機能
- フローチャート

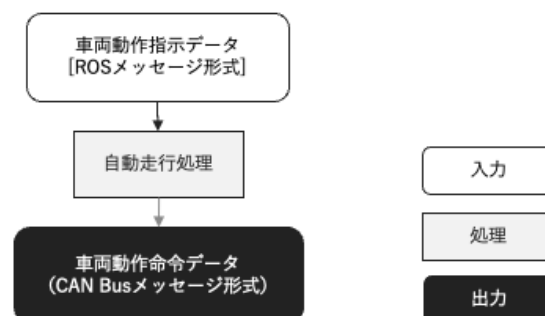


図 4-47 自動走行処理

- データ仕様
 - 入力
 - ◇ 車両動作指示データ（ROS メッセージ形式）
 - 内容
 - 車両の動作指示を表したデータ
 - 形式

- ROS メッセージ
- データ詳細
 - 内部連携インタフェース【IF265】を参照
- 出力
 - ◇ 車両動作命令データ（CAN Bus メッセージ形式）
 - 内容
 - 車両の動作命令を表したデータ
 - 形式
 - CAN Bus 形式
 - データ詳細
 - 内部連携インタフェース【IF265】を参照
- 機能詳細
 - 車両動作指示
 - ◇ 昨年度からの改善点
 - 昨年度、走行はせずに、検証用の車両を使ったデータ採取と自己位置測位のみに留まったが、本年度は実際に電動トラックを走行させる機能を実装する
 - ◇ 処理内容
 - Autoware から送られる車両の動作指示を基に、車両へ CAN Bus 形式の動作命令を送る
 - ◇ 利用するライブラリ
 - python-can（ソフトウェア・ライブラリ【SL115】を参照）
 - ◇ 利用するアルゴリズム
 - なし

27. 【FN107】 ルート設定

- 機能概要
 - 車両が走行するウェイポイントを設定する機能
- フローチャート

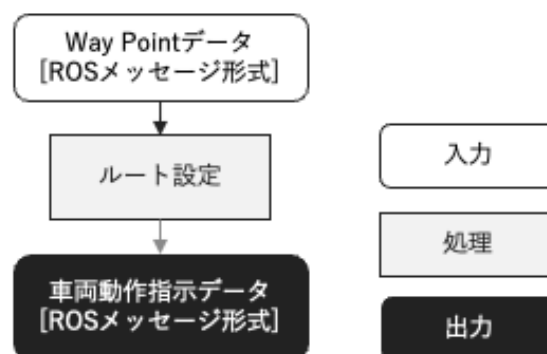


図 4-48 ルート設定

- データ仕様
 - 入力
 - ✧ ウェイポイントを Autoware で設定
 - 内容
 - 車両が走行するルートを設定
 - 形式
 - なし
 - データ詳細
 - 内部連携インタフェース【IF266】を参照
 - 出力
 - ✧ 車両動作指示データ（ROS メッセージ形式）
 - 内容
 - 車両の動作指示を表したデータ
 - 形式
 - ROS メッセージ
 - データ詳細
 - 内部連携インタフェース【IF265】を参照
 - 機能詳細
 - ルート設定
 - ✧ 処理内容
 - 実車両を走行させるため、点群マップが読み込まれた仮想空間内にウェイポイントを設定する機能
 - ✧ 利用するライブラリ
 - Autoware-planning（ソフトウェア・ライブラリ【SL115】を参照）
 - ✧ 利用するアルゴリズム
 - なし
28. 【FN108】データ送信
- 機能概要
 - 車両のテレメトリデータを SENSYN CORE Pilot へ送信する機能
 - フローチャート

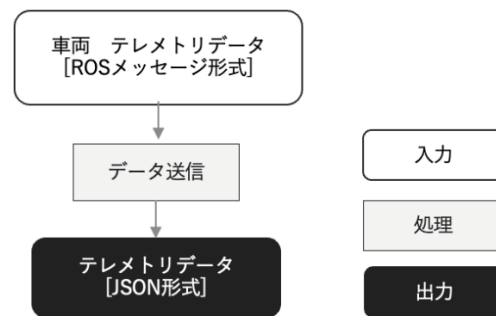


図 4-49 データ送信

- データ仕様
 - 入力
 - ✧ 車両テレメトリデータ
 - 内容
 - 車両テレメトリデータの ROS メッセージ
 - 形式
 - ROS メッセージ
 - データ詳細
 - 内部連携インタフェース【IF264】を参照
 - 出力
 - ✧ MQTT メッセージ
 - 内容
 - 車両テレメトリデータが含まれている JSON 形式のメッセージ
 - 形式
 - JSON 形式
 - データ詳細
 - 内部連携インタフェース【IF207】を参照
- 機能詳細
 - データ送信
 - ✧ 処理内容
 - 車両テレメトリデータを SENSYN CORE Pilot へ送信する
 - ✧ 利用するライブラリ
 - mqtt_bridge (ソフトウェア・ライブラリ【SL109】を参照)
 - ✧ 利用するアルゴリズム
 - なし

29. 【FN109】 データ受信

- 機能概要
 - SENSYN CORE Pilot から送られる車両停止指示データや走行ルートを受信する機能
- フローチャート

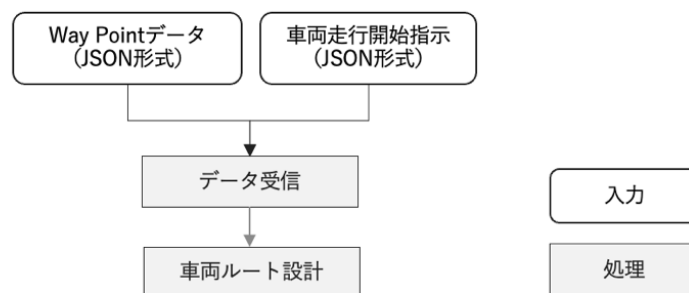


図 4-50 データ受信

- データ仕様
 - 入力
 - ◇ ウェイポイントデータ
 - 内容
 - ウェイポイントデータが含まれている JSON 形式のメッセージを SENSYN CORE Pilot から受信
 - 形式
 - JSON 形式
 - データ詳細
 - 内部連携インタフェース【IF208】を参照
 - ◇ 走行開始データ
 - 内容
 - 走行開始指示が含まれている JSON 形式のメッセージを SENSYN CORE Pilot から受信
 - 形式
 - JSON 形式
 - データ詳細
 - 内部連携インタフェース【IF208】を参照
 - 出力
 - なし
- 機能詳細
 - データ受信
 - ◇ 処理内容
 - ウェイポイントデータ、走行開始指示を SENSYN CORE Pilot から受信する

- ✧ 利用するライブラリ
 - mqtt_bridge（ソフトウェア・ライブラリ【SL109】を参照）
- ✧ 利用するアルゴリズム
 - なし

30. 【FN110】データ受信・配信

- 機能概要
 - 車両と SENSYN CORE Pilot 間のデータ受信・配信を MQTT メッセージで送受信するためにサーバーに設置される機能
- フローチャート

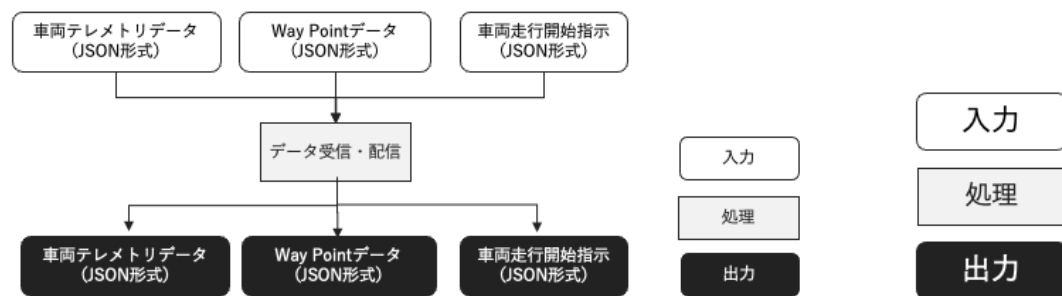


図 4-51 データ受信・配信

- データ仕様
 - 入力
 - ✧ 車両テレメトリデータ
 - 内容
 - 測位データが含まれている JSON 形式のメッセージを車両から受信
 - 形式
 - JSON 形式
 - データ詳細
 - 内部連携インタフェース【IF207】を参照
 - 出力
 - ✧ ウェイポイントデータ
 - 内容
 - ウェイポイントデータが含まれている JSON 形式のメッセージを SENSYN CORE Pilot から受信
 - 形式
 - JSON 形式
 - データ詳細
 - 内部連携インタフェース【IF208】を参照
 - ✧ 車両走行開始データ
 - 内容

- 走行開始指示が含まれている JSON 形式のメッセージを SENSYN CORE Pilot から受信
- 形式
 - JSON 形式
- データ詳細
 - 内部連携インタフェース【IF208】を参照
- 機能詳細
 - データ受信・配信
 - ✧ 処理内容
 - 車両から受信したデータを SENSYN CORE Pilot へ配信する
 - ✧ 利用するライブラリ
 - mqtt_bridge（ソフトウェア・ライブラリ【SL109】を参照）
 - ✧ 利用するアルゴリズム
 - なし

4-3. アルゴリズム

4-3-1. 利用したアルゴリズム

- ドローン用アルゴリズム

表 4- 6 利用したアルゴリズム一覧

ID	アルゴリズムを利用した機能	名称	説明	選定理由
AL001	FN004 FN005 FN105	NDT Registration	● 異なる二つの点群間の位置合わせに相応しい移動・回転量を見つけるためのアルゴリズム ● 事前に準備した点群地図と LiDAR から得られるスキャンを位置合わせすることで点群地図における LiDAR の位置を算出する	● 高速にスキャン同士のマッチングが可能であり、自律移動のようなリアルタイムアプリケーションに適しているため ● 大規模な地図データを用いても高速に動作するため
AL002	FN009	FAST アルゴリズム	● 画像中からコーナー特徴量を検出するアルゴリズムで、非常に高速に動作するアルゴリズム	● リソースの限られたデバイスなどでも実行が容易で、ドローンや移動ロボットなどのリアルタイム性が要求されるアプリケーションに適しているため
AL003	FN009 FN010	拡張カルマンフィルタ： EKF (Extended Kalman Filter)	● 複数の測位情報の統合を行うことで真値に近くノイズに強い結果を得ることができるアルゴリズム	● 他の情報統合アルゴリズムなどに比べて高速に実行することができるため
AL004	FN017	辞書ベースのデータマッピングアルゴリズム	● 複数システム間でのプロトコル変換を行うアルゴリズム ● ROS メッセージで表現されたテレメトリー情	● 多種類のデータを高速にマッピングするのに適しているため ● 状態や順番を考慮せず利用できるアルゴリズムであるため

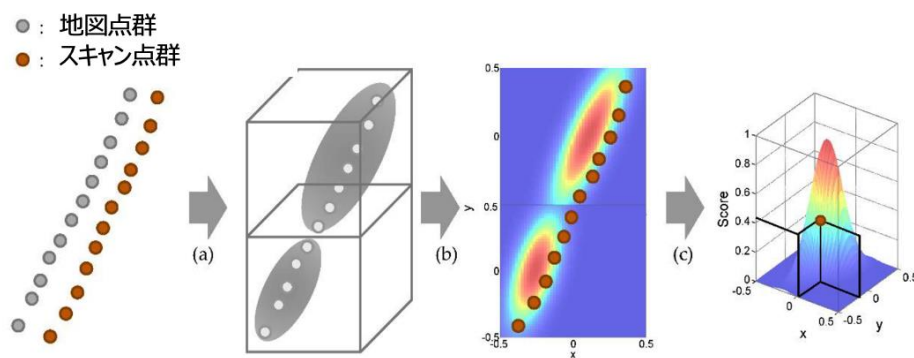
			報を MAVLink に変換するために利用する	
AL005	FN017	State machine ベースの変換アルゴリズム	● 複数システム間でのプロトコル変換を行うアルゴリズム ● MAVLink で表現されている飛行ミッションを ROS メッセージに変換するために利用	● 辞書ベースのマッピングではできない、複数のメッセージをまとめるため ● 状態や順番を考慮できるアルゴリズムであるため

1) 【AL001】 NDT Registration

● 計算量

- 地図点群が持つ点の数を M としたとき地図点群をボクセルに分割し各セル内のポイントから正規分布を割り出すための計算量は $O(M)$
- スキャン点群が持つ点の数を N とし、ボクセルの数を v としたとき、また反復的な最適化が行われる回数が κ であったときの計算量は $O(\kappa C)$ となる。このとき $C = Nv$
- また反復ごとに収束したかチェックを行うがこのときの計算量は $O(1)$
- 全体を通しての NDT Registration の計算量は $O(M + \kappa C)$ となる

● イメージ

図 4-52 NDT Registration による各点が持つ一致度評価のイメージ³

2) 【AL002】 FAST アルゴリズム

● 計算量

- 画像中のコーナーをターゲットとして検出する高速動作する特徴量検出アルゴリズム。画素周辺の

³ 出典：A Review of Point Cloud Registration Algorithms for Laser Scanners: Applications in Large-Scale Aircraft Measurement (<https://www.mdpi.com/2076-3417/12/20/10247>)

明暗さのパターンによりコーナーを検出する

➤ 画像サイズが $N \times M$ である場合、計算量は $O(NM)$

● イメージ

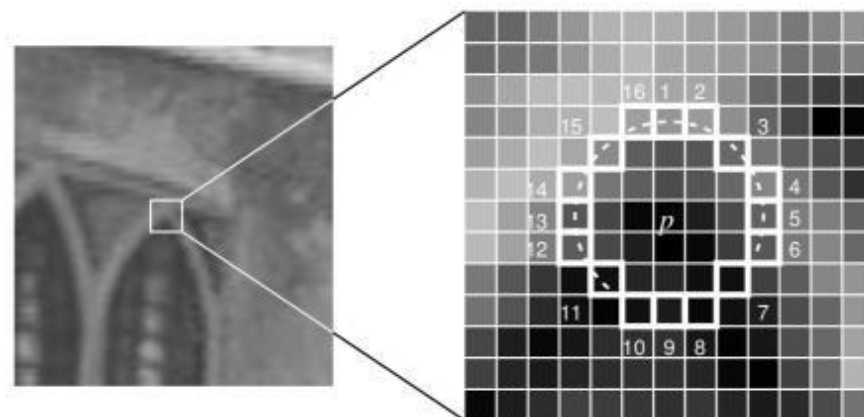


図 4-53 FAST アルゴリズムによるビジュアルオドメトリのイメージ⁴

3) 【AL003】 拡張カルマンフィルタ：EKF (Extended Kalman Filter)

● 計算量

- ノイズを含む複数の情報を統合し、センサーの弱点を補い合うことでより真値に近い情報へとフィルタリングを行う
- 状態ベクトルの次元数を N としたとき $O(N^3)$

● イメージ

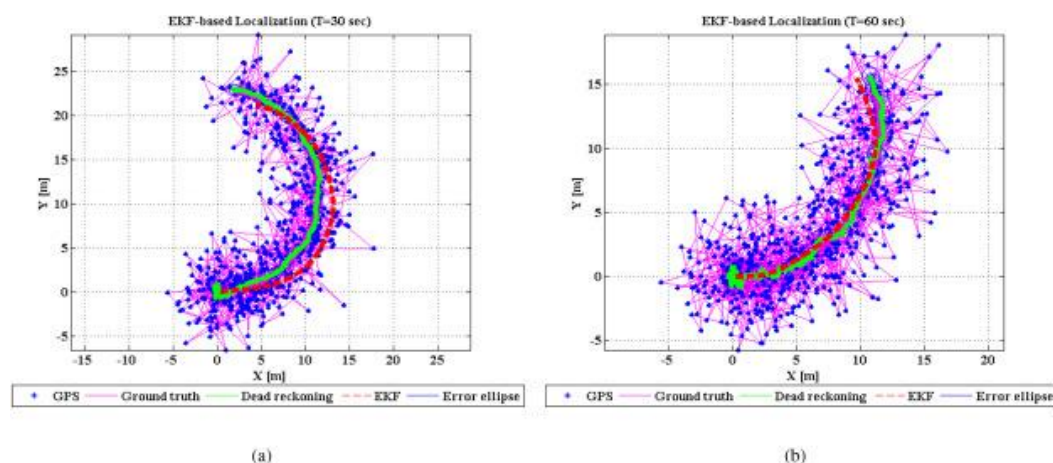


図 4-54 EKF(Extended Kalman Filter)による自己位置推定軌道（赤色点）のイメージ⁵

⁴ 出典：Edward Rosten and Tom Drummond, “Machine learning for high speed corner detection” in 9th European Conference on Computer Vision, vol. 1, 2006, pp. 430–443.

⁵ 出典：Extended Kalman Filter-based localization algorithm by edge computing in Wireless Sensor Networks (<https://www.sciencedirect.com/science/article/pii/S2352864820302601#fig5>)

4) 【AL004】辞書ベースのデータマッピングアルゴリズム

- 計算量

- ある ROS メッセージについて、辞書を参照し 1 対 1 で対応する Mavlink メッセージを特定し、データマッピングを行う（逆方向の Mavlink メッセージ to ROS メッセージについても同様）
- 1 対 1 で対応する相手方メッセージを検索するため、その計算量は $O(1)$
- 変換を行うメッセージの数を n としたとき $O(n)$

- イメージ

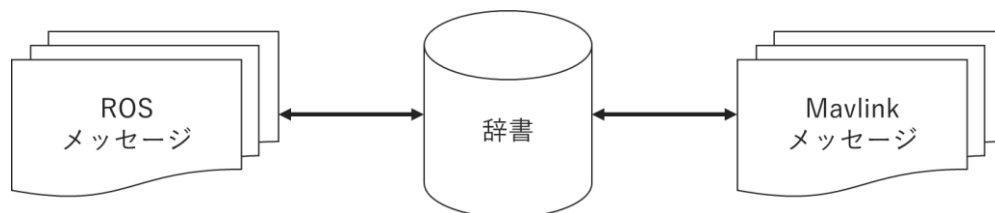


図 4-55 辞書ベースのデータマッピングアルゴリズムによる ROS メッセージ変換のイメージ

5) 【AL005】State machine ベースの変換アルゴリズム

- 計算量

- あるウェイポイント情報を構築するために必要な各種情報が複数の Mavlink メッセージで送られてくるため、そのメッセージを受信し変換、統合することで 1 つのウェイポイント情報を作成する
- 1 つのウェイポイントを作成する際に必要な Mavlink メッセージの数を n としたとき $O(n)$

- イメージ

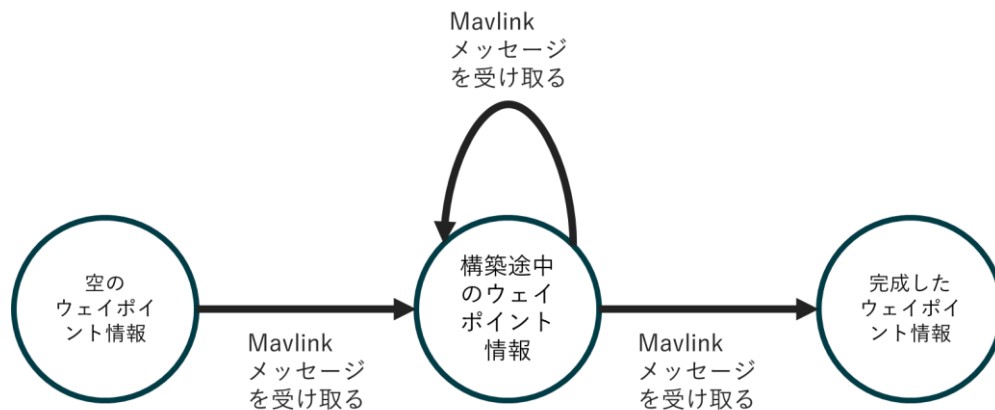


図 4-56 State machine ベースの変換アルゴリズムによる ROS メッセージ変換のイメージ

4-3-2. 開発したアルゴリズム

1) 【AL101】 ビジュアルイナーシャルオドメトリ (VIO)

- 本アルゴリズムを利用した機能

- 【FN009】 ビジュアルイナーシャルオドメトリ (VIO)

- アルゴリズムの詳細

ドローン自律運航システムのロバスト性の更なる向上を目的として、LiDAR と GNSS (衛星測位システム) による既存の自己位置推定技術に加え、新たにビジュアルイナーシャルオドメトリ (VIO) の開発を行った。ビジュアルイナーシャルオドメトリ (VIO) とは画像中からコーナー特徴量を検出するアルゴリズムで、非常に高速に動作する FAST アルゴリズム (AL002) により抽出した各画像に対する特徴点と IMU 情報により、局所的なカメラの移動量を推定するアルゴリズムである。

先行研究として、Anand George "Analysis of Visual-Inertial Odometry Algorithms for Outdoor Drone Applications"⁶などが挙げられる。また、FAST アルゴリズムのほかに、SIFT アルゴリズムは、スケール不偏性を持ち、異なる距離やズーム状態で取得した画像間でも特徴量の精密なマッチングが可能であることから、速度と精度のバランスが良いと言われている。SIFT に近い高精度な特徴量抽出が可能であり、計算速度が比較的高速な AKAZE アルゴリズムなどが存在するが、本ユースケースではドローン飛行を行うため、ドローン飛行に必要なリアルタイム性を満たす、高速な FAST アルゴリズムを採用した。

ステップ

1. カメラにより取得した単一画像に対しコーナー特徴量を検出する。この際 FAST アルゴリズム (AL002) を用いる。あるピクセルを p としそのピクセルの値を I_p としたとき、円形の周囲 16 ピクセルを調査する。もし調査したピクセルの値が 10 個連続して I_p より低い又は連続して値が高い場合は画素 p をコーナーとして検出する。
2. 前回画像フレームで得た特徴量と新しく得られた画像フレーム内の特徴量の対応関係を求める。具体的には前回得た特徴量の周囲数ピクセルの変化を勾配として捉え、各特徴点周辺の移動量を算出し、画像全体としての画像内での平行移動、回転を決定する。また、特徴量の新しい位置を決定する。
3. マッチングで特定した特徴点のペアを用い、カメラの相対的な姿勢や移動量 (平行移動・回転量) を求める。このとき 5 点以上の対応点を用い基本行列(Essential matrix)を最適化により求める。ただしこの際実際の距離に関しては不確定となる。
4. 積算した IMU の加速度と角速度情報を積算することにより、画像フレーム間での実際の移動量を算出。特徴量追跡により得られた平行移動量・回転量に対してスケールの調整を行い正確な移動量を出す。

⁶ 出典：<https://aaltodoc.aalto.fi/server/api/core/bitstreams/a0655bd9-d0bd-45ec-a57b-f709f43fe0d8/content>

各ステップの
処理イメージ

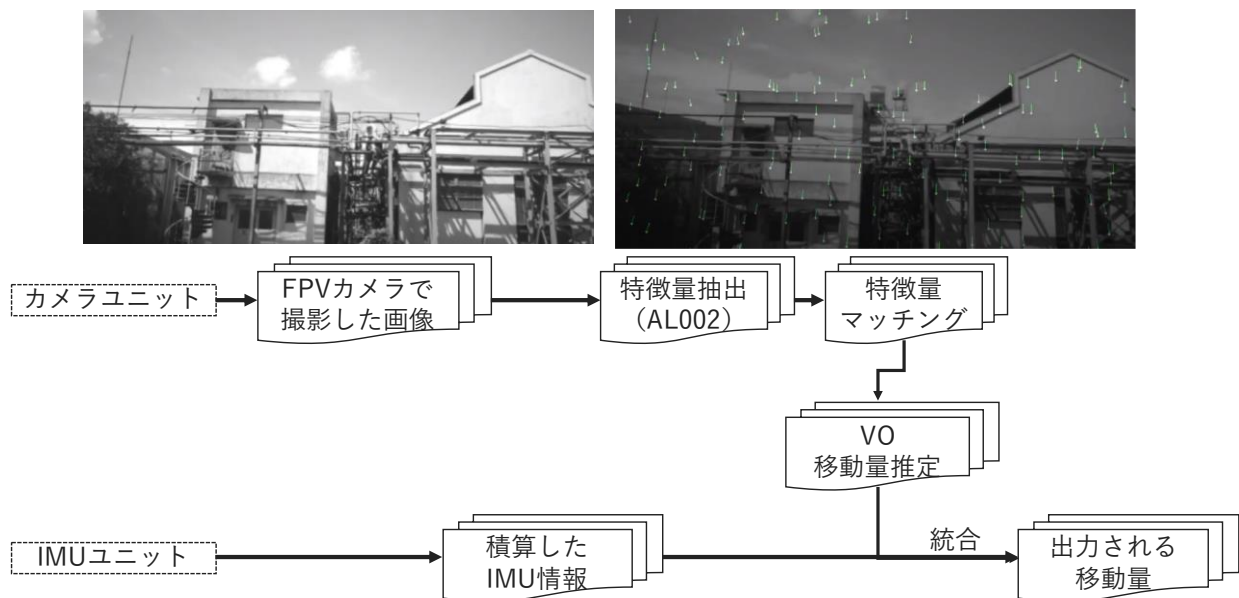


図 4-57 アルゴリズム詳細

2) 【AL102】測位情報の統合アルゴリズム

- 本アルゴリズムを利用した機能
 - 【FN010】測位情報統合
 - 【FN011】障害物回避
 - 【FN012】目標速度算出

● アルゴリズムの詳細

測位情報・センサー情報の統合・取捨選択を行うアルゴリズムを開発した。

具体的には GPS、LiDAR 自己位置推定からの位置情報、IMU からの加速度・角速度、VIO からの移動速度など複数の異なる観測情報を統合し、各センサー単体よりもロバストな自己位置推定を実現する。また、統合される直前で測位情報に対し信頼できるかどうかの検査が行われ信頼できる場合にのみ統合に用いる技術も用いられており、システムの堅牢性向上に寄与している。

1. 運動モデル定義： ロボットの動きを記述するために定義されるシステムモデルを定義する。ドローンの運動モデルは 3 次元位置と 3 次元姿勢を考慮するため、6 自由度。このときの状態ベクトルは以下のとおりである。

$[X, Y, Z, roll, pitch, yaw, \dot{X}, \dot{Y}, \dot{Z}, \dot{roll}, \dot{pitch}, \dot{yaw}, \ddot{X}, \ddot{Y}, \ddot{Z}]$

2. 観測モデル定義： システムの現在の状態（状態ベクトル）から、センサーが測定するであろう情報（観測データ）を予測する方程式を決める。このときの観測ベクトルは以下のとおりである。

$[x_{\text{liear}k}, y_{\text{liear}k}, z_{\text{liear}k}, r_{\text{liear}k}, p_{\text{liear}k}, y_{\text{liear}k}, x_{\text{pos}k}, y_{\text{pos}k}, z_{\text{pos}k}, \omega_{\text{roll}k}, \omega_{\text{pitch}k}, \omega_{\text{yaw}k}, x_k, a_{yk}, a_{zk}]$

3. 予測ステップ： 運動モデルに基づいて、現在の状態ベクトルから次の時刻の状態ベクトルを予測。過去の観測データから現在又は未来の自己位置などを予測し利用する。更新式は以下のとおりである。

$$x_t = f(x_{t-1}, u_{t-1})$$

x_t は時刻 t における状態ベクトル

u_t は時刻 t における制御入力. ただし $u_t=0$

4. 更新ステップ： 観測データを用いて、状態変数及び共分散行列の予測値修正を更新ステップで行う。この際各センサーや測位情報が入力されるが、使用される測位情報の信頼度が低い場合は信頼度の低い情報は無視される。状態ベクトルの更新は以下のように行われる。

$$y_t = z_t - H_t x_t|_{t-1}$$

z_t は時刻 t の実際の観測値

z_t は時刻 t の実際の観測値

H_t は時刻 t での観測モデルのヤコビ行列

5. 3 と 4 を繰り返す

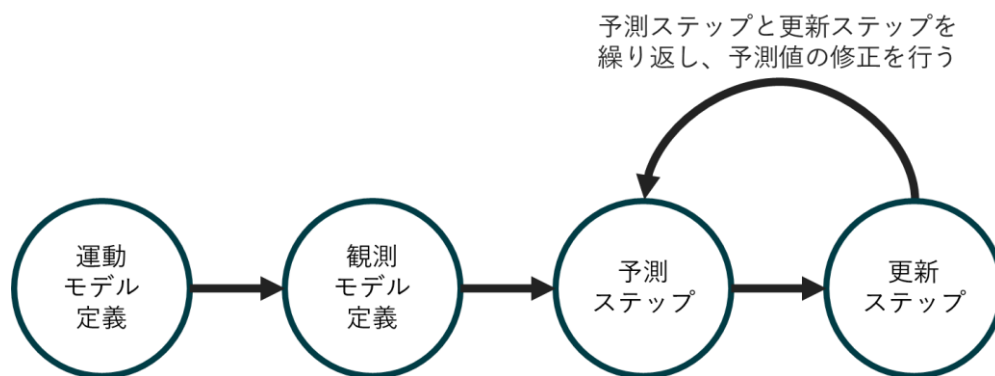


図 4-58 アルゴリズム概要

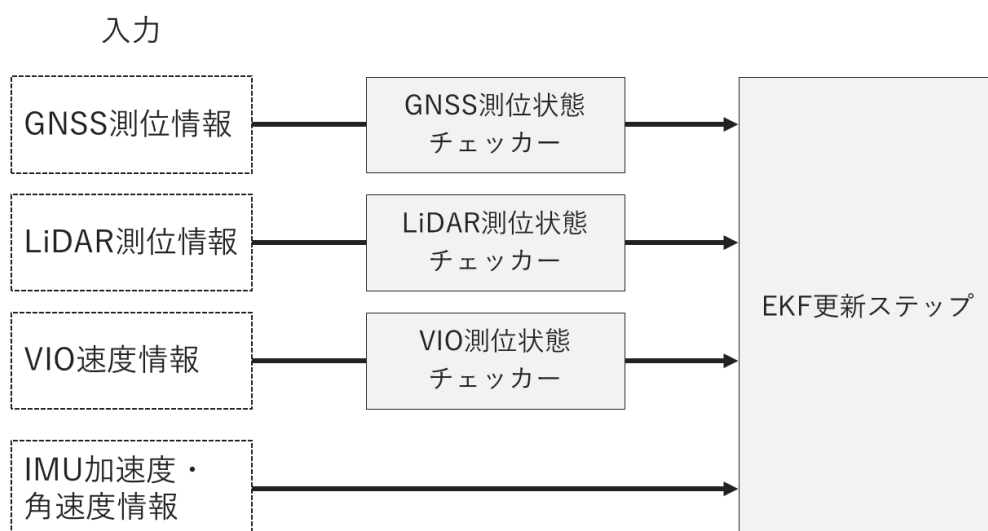


図 4-59 EKF 更新ステップ入力概要

- 3) 【AL103】 日本国内を囲む緯度・経度座標のリストを利用し、指定の緯度・経度座標が囲まれた中に存在するかを判定するアルゴリズム

- イメージ

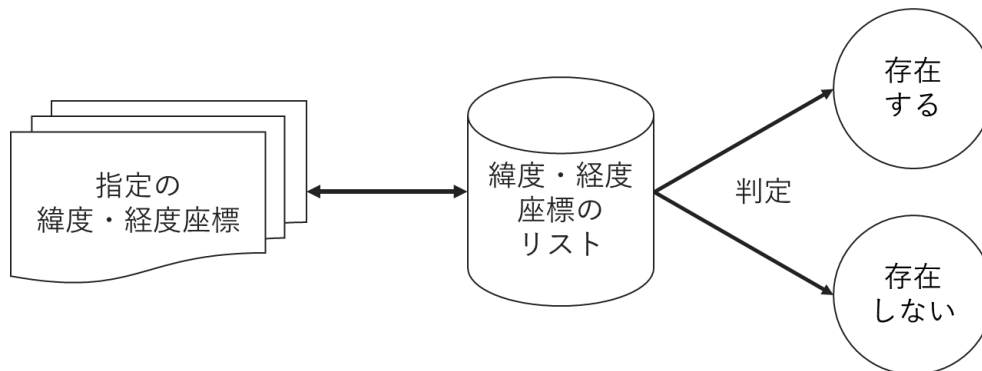


図 4-60 日本国内を囲む緯度・経度座標のリストを利用し、指定の緯度・経度座標が囲まれた中に存在するかを判定するアルゴリズムによる判定のイメージ

- 本アルゴリズムを利用した機能

- 【FN014】 地形データ切り替え

- アルゴリズムの詳細

指定の緯度・経度座標が囲まれた中に存在するかを判定するアルゴリズム

- 4) 【AL104】 PLATEAU-Terrain を用いるルートかどうかを判定するアルゴリズム

- イメージ

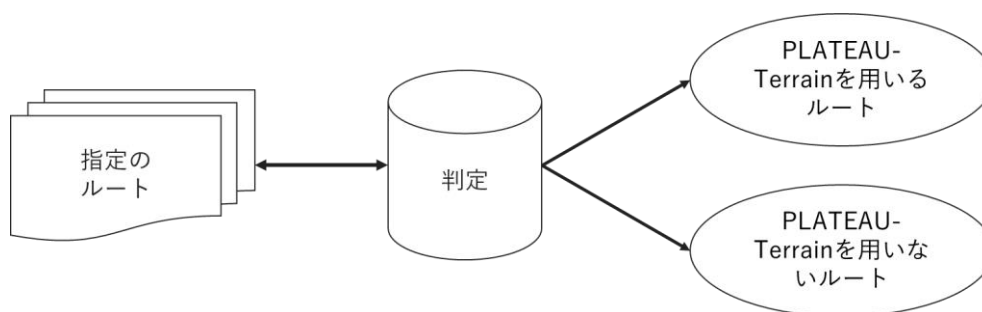


図 4-61 PLATEAU-Terrain を用いるルートかどうかを判定するアルゴリズムによる判定のイメージ

- 本アルゴリズムを利用した機能

- 【FN014】 地形データ切り替え

- アルゴリズムの詳細

SENSYN CORE Pilot で作成した飛行ルートについて、PLATEAU-Terrain を用いるルートかどうかを判定するアルゴリズム

4-4. データインタフェース

4-4-1. ファイル入力インタフェース

1) 【IF001】 3D 都市モデル CityGML ファイル入力

- 本インタフェースを利用する機能：【FN001】

表 4-7 3D 都市モデル CityGML ファイル入力

入力ファイル名称	形式	詳細
51357380_bldg_6697_op.gml	CityGML 形式	大阪市此花区夢洲地区（実証エリア）の都市設備 LOD3 データと大阪・関西万博パビリオンの建築物 LOD3～4 データをマージした 3D 都市モデル

4-4-2. ファイル出力インタフェース

1) 【IF101】 3D 都市モデルの点群変換 PCD ファイル出力

- 本インタフェースを利用する機能
 - 【FN001】

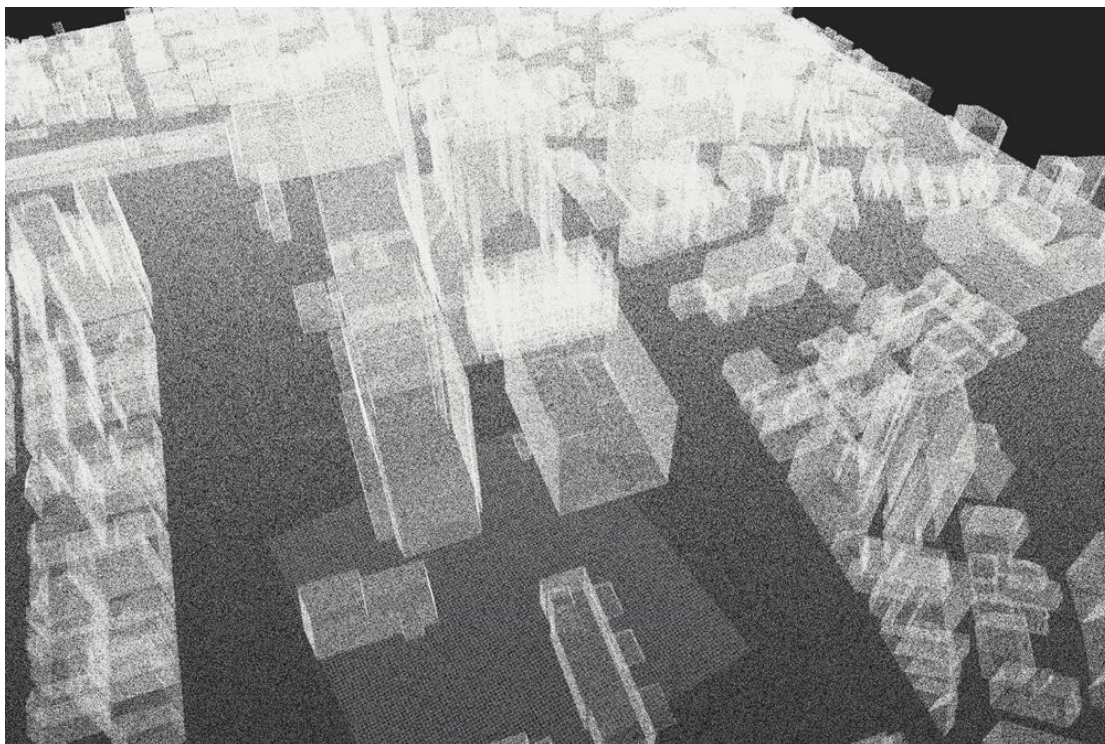


図 4-62 3D 都市モデルの点群変換 PCD ファイル出力で出力した PCD ファイルイメージ図

表 4-8 3D 都市モデルの点群変換 PCD ファイル出力

出力ファイル名称	形式	詳細
elas_map.pcd	PCD 形式	大阪市此花区夢洲地区（実証エリア）の都市設備 LOD3 データと大阪万博パビリオンの建築物 LOD3～4 データをマージした 3D 都市モデルを、PCD 形式に変更したもの。 PCD 形式に変換する際、各建物についてそれぞれ一番 LOD のレベルが高いデータを点群化する。これにより、建築物 LOD4 が唯一存在する大阪万博パビリオンは LOD4 をベースに、他の建築物は LOD3 をベースとした点群データ（pcd ファイル）が出力される。

2) 【IF102】 PLATEAU PCL Generator PCD ファイル出力

- 本インターフェースを利用する機能
 - 【FN101】

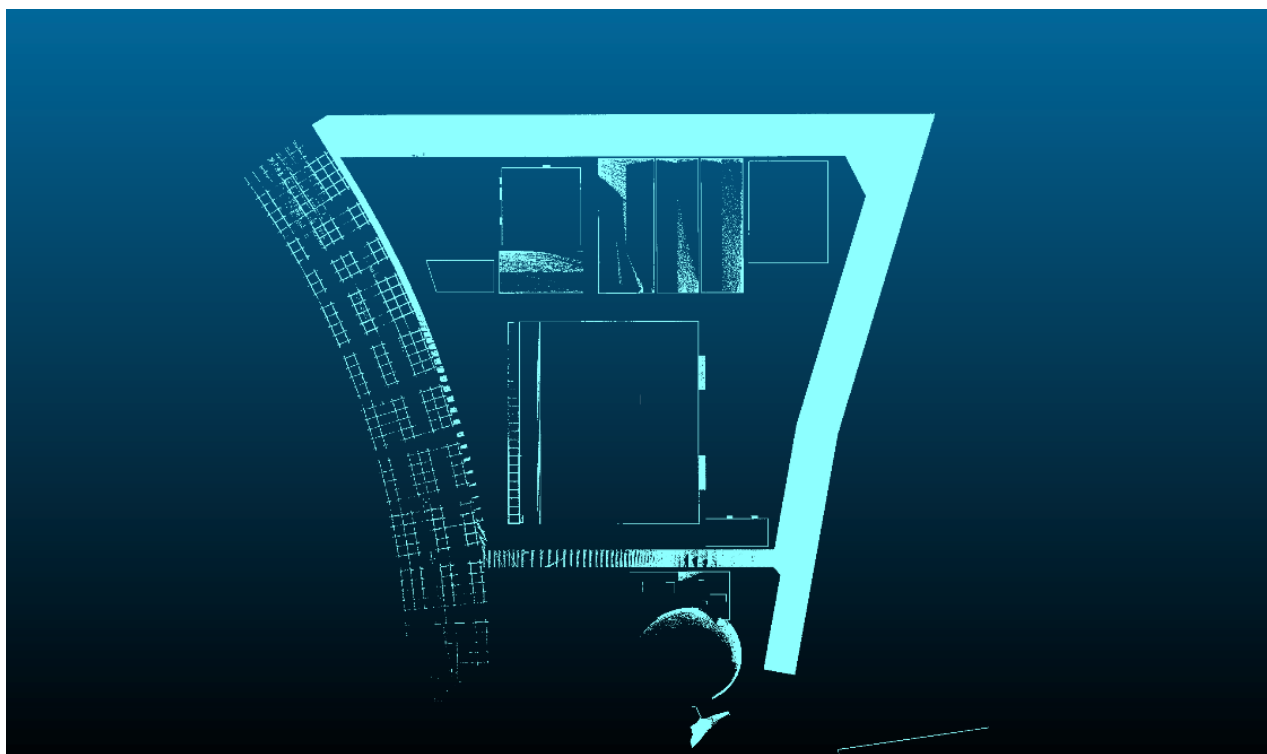


図 4-63 PLATEAU PCL Generator PCD ファイル出力 イメージ図

表 4-9 PLATEAU PCL Generator PCD ファイル出力

出力ファイル名称	形式	詳細
pclg_map.pcd	PCD 形式	走行エリアの CityGML ファイルを PCLG で読み込み、車両を仮想空間内で走行させることで出力される点群マップ。

4-4-3. 内部連携インターフェース

データの入出力は ROS メッセージ等により直接行う。

1) 【IF201】 LiDAR スキャンデータの連携

- インタフェースの詳細
 - LiDAR ユニットで取得したスキャンデータを ELAS 側で受け取る
- 本インタフェースを利用する機能
 - 【FN004】 【FN005】 【FN006】 【FN011】
- プロトコル
 - ROS メッセージ
- パス
 - / fusion/ lidar
- メッセージ型
 - AnswerForm

表 4-10 メッセージ型 (AnswerForm)

フィールド	説明	値
lidar_scan	LiDAR スキャンデータ	-

表 4-11 LiDAR スキャンデータの連携

名称	形式	タイミング・頻度
LiDAR スキャンデータ	Point Cloud 形式	20 Hz (1 秒間に 20 回取得)

2) 【IF202】 GNSS 測位座標の連携

- インタフェースの詳細
 - GNSS ユニットで取得した測位情報を ELAS 側で受け取る
- 本インタフェースを利用する機能
 - 【FN007】 【FN008】 【FN010】
- プロトコル
 - ROS メッセージ
- パス
 - / fusion/ gnss
- メッセージ型
 - AnswerForm

表 4-12 メッセージ型 (AnswerForm)

フィールド	説明	値
gnss	GNSS 測位座標データ	-

表 4-13 GNSS 測位座標 データの連携

名称	形式	タイミング・頻度
GNSS 測位座標	NMEA 形式	5 Hz (1 秒間に 5 回取得)

3) 【IF203】カメラ画像データの連携

- インタフェースの詳細
 - カメラユニットで取得した画像を ELAS 側で受け取る
- 本インタフェースを利用する機能
 - 【FN009】
- プロトコル
 - ROS メッセージ
- パス
 - / fusion/ camera
- メッセージ型
 - AnswerForm

表 4-14 メッセージ型 (AnswerForm)

フィールド	説明	値
fpv_img	カメラ映像データ	-

表 4-15 カメラ映像データの連携

名称	形式	タイミング・頻度
カメラ映像データ	Image 形式	30 Hz (1 秒間に 30 回取得)

4) 【IF204】IMU (機体) 加速度/角速度データの連携

- インタフェースの詳細
 - IMU ユニットで取得した加速度/角速度を ELAS 側で受け取る
- 本インタフェースを利用する機能
 - 【FN009】 【FN010】 【FN012】
- プロトコル
 - ROS メッセージ
- パス
 - / fusion/ imu
- メッセージ型
 - AnswerForm

表 4-16 メッセージ型 (AnswerForm)

フィールド	説明	値
acceleration	加速度	-
angular_velocity	角速度	-

表 4-17 IMU (機体) 加速度/角速度データの連携

名称	形式	タイミング・頻度
IMU (機体) 加速度/角速度	ROS メッセージ形式	200 Hz (1 秒間に 200 回取得)

5) 【IF205】 ELAS、SENSYN CORE Pilot 間連携インタフェース (ROS メッセージ形式)

- 本インタフェースを利用する機能
 - 【FN003】 【FN011】 【FN012】 【FN013】 【FN015】 【FN017】 【FN018】 【FN019】 【FN020】
- プロトコル
 - ROS メッセージ

表 4-18 ELAS、SENSYN CORE Pilot 間連携インタフェース (ROS メッセージ形式)

#	インタフェース名	内容
1	目標経路 (飛行ルート) の送信	● 飛行ルートを送信する ➢ 実行する飛行経路のウェイポイント (緯度・経度・高度) ➢ 各ウェイポイントにおけるドローンの速度 ➢ 飛行ルートの送信結果 (送信成功/失敗) 表示
2	飛行開始指示	● 機体に送信された飛行ルートでのミッションを開始し、離陸する指示を送信する
3	緊急停止指示	● 進行中のミッションを一時停止し、その場でホバリングする指示を送信する
4	飛行再開指示	● 一時停止した、進行中のミッションを再開する
5	テレメトリー情報の表示	● 機体の各種テレメトリー情報を SENSYN CORE Pilot に送信する ➢ 機体名、カメラ名 ➢ 機体の接続状況 ➢ 機体の位置 (緯度・経度・高度) ➢ 機体の進行速度 (m/s) ➢ 機体の自己位置推定ステータス (GNSS 測位状態、LiDAR 測位状態) ➢ 機体のバッテリー残量 ➢ 飛行モード ➢ 飛行予定時間

6) 【IF206】自己位置推定情報/ドローン移動量推定情報の連携

- 本インターフェースを利用する機能
 - 【FN006】 【FN009】 【FN010】

表 4-19 ELAS、SENSYN CORE Pilot 間連携インターフェース

#	インターフェース名	内容
1	LiDAR 自己位置推定情報の連携	● LiDAR ローカリゼーションにより出力された、LiDAR の自己位置推定情報を【FN006】 LiDAR 測位情報座標変換機能で基準座標系に直したデータを【FN010】 測位情報統合に連携する
2	ドローン移動量推定情報の連携	● 【FN009】 ビジュアルイナーシャルオドメトリ (VIO) より得られるドローン移動量推定値 (xyz 速度、角速度推定値) を【FN010】 測位情報統合に連携する

7) 【IF207】車両、SENSYN CORE Pilot 間連携インターフェース

- 本インターフェースを利用する機能
 - 【FN018】 【FN108】
- プロトコル
 - MQTT メッセージ

表 4-20 車両、SENSYN CORE Pilot 間連携インターフェース

#	インターフェース名	内容
1	テレメトリー情報の表示	● 機体の各種テレメトリー情報を SENSYN CORE Pilot に送信する ➤ 機体名、カメラ名 ➤ 機体の接続状況 ➤ 機体の位置 (緯度・経度) ➤ 機体の進行速度 (m/s) ➤ 機体の自己位置推定ステータス (GNSS 測位状態) ➤ 機体のバッテリー残量

8) 【IF208】SENSYN CORE Pilot から車両ヘルート情報の連携インターフェース

- 本インターフェースを利用する機能
 - 【FN109】
- プロトコル
 - MQTT メッセージ

表 4-2 1 車両、SENSYN CORE Pilot 間連携インタフェース

#	インタフェース名	内容
1	目標経路（飛行ルート）の送信	- 走行ルートを送信する ➤ 実行する走行経路のウェイポイント（緯度・経度） ➤ 各ウェイポイントにおける車両の速度 ➤ 飛行ルートの送信結果（送信成功/失敗）表示
2	走行開始指示	機体に送信された走行ルートでのミッションを開始し、走行開始する指示を送信する

9) 【IF209】ELAS、SENSYN CORE Pilot 間連携インタフェース（MAVLink 形式）

- 本インタフェースを利用する機能
 - 【FN015】 【FN017】 【FN020】
- プロトコル
 - MAVLink 形式

表 4-2 2 ELAS、SENSYN CORE Pilot 間連携インタフェース(MAVLink 形式)

#	インタフェース名	内容
1	目標経路（飛行ルート）の送信	- 飛行ルートを送信する ➤ 実行する飛行経路のウェイポイント（緯度・経度・高度） ➤ 各ウェイポイントにおけるドローンの速度 ➤ 飛行ルートの送信結果（送信成功/失敗）表示
2	飛行開始指示	機体に送信された飛行ルートでのミッションを開始し、離陸する指示を送信する
3	緊急停止指示	進行中のミッションを一時停止し、その場でホバリングする指示を送信する
4	飛行再開指示	一時停止した、進行中のミッションを再開する
5	テレメトリ情報の表示	- 機体の各種テレメトリ情報を SENSYN CORE Pilot に送信する ➤ 機体名、カメラ名 ➤ 機体の接続状況 ➤ 機体の位置（緯度・経度・高度） ➤ 機体の進行速度（m/s） ➤ 機体の自己位置推定ステータス（GNSS 測位状態、LiDAR 測位状態） ➤ 機体のバッテリー残量 ➤ 飛行モード

		➤ 飛行予定時間
--	--	----------

10) 【IF251】 PLATEAU PCL Generator インタフェース

- 本インタフェースを利用する機能
 - 【FN101】
- プロトコル
 - なし

表 4-2 3 PLATEAU PCL Generator インタフェース

#	インタフェース名	内容
1	PLATEAU SDK for Unity	● 3D 都市モデルを Unity に読み込む ➤ CityGML の選択 ➤ 読み込みエリアの設定 ➤ LOD の設定
2	点群設定	● PCLG 上の LiDAR 設定を行う ➤ 放射角度の設定 ➤ 飛距離の設定
3	ルート設定	● ルート設定を行う ➤ ウェイポイントの設置

11) 【IF261】 点群データ読み込みインタフェース

- 本インタフェースを利用する機能
 - 【FN102】 【FN105】
- プロトコル
 - ROS メッセージ

表 4-2 4 点群データ読み込みの連携インタフェース

#	インタフェース名	内容
1	LiDAR データ	● 3D LiDAR から読み込まれるデータを ROS メッセージに変換する ➤ 点群座標 ➤ 点群強度

12) 【IF262】 NMEA データ読み込みインタフェース

- 本インターフェースを利用する機能
 - 【FN103】【FN104】【FN105】
- プロトコル
 - ROS メッセージ

表 4-2 5 NMEA データ読み込みインターフェース

#	インターフェース名	内容
1	NMEA データ	● RTK-GNSS 測位デバイスから読み込まれるデータを ROS メッセージに変換する ➤ 緯度経度情報

13) 【IF263】センサーデータ処理インターフェース

- 本インターフェースを利用する機能
 - 【FN104】
- プロトコル
 - ROS メッセージ

表 4-2 6 NMEA データ読み込みインターフェース

#	インターフェース名	内容
1	LiDAR データ	● LiDAR から得られる点群データに対し、間引き処理やノイズ除去を行う
2	NMEA データ	● RTK 測位デバイスから得られる緯度経度情報に対し、座標変換などの処理を行う

14) 【IF264】自己位置処理インターフェース

- 本インターフェースを利用する機能
 - 【FN105】
- プロトコル
 - ROS メッセージ

表 4-2 7 自己位置処理インターフェース

#	インターフェース名	内容
1	自己位置データ	● Autoware 座標系で得られる車両の位置と座標を計算する

15) 【IF265】自動走行処理インターフェース

- 本インターフェースを利用する機能
 - 【FN105】 【FN106】 【FN107】
- プロトコル
 - CAN-Bus

表 4-28 自己位置処理インターフェース

#	インターフェース名	内容
1	Can-Bus	● 動作命令を基に、車両へ走行指示を出す

16) 【IF266】 ルート設定インターフェース

- 本インターフェースを利用する機能
 - 【FN107】
- プロトコル
 - ROS メッセージ

表 4-29 自己位置処理インターフェース

#	インターフェース名	内容
1	ルート情報	● 車両の走行開始地点とルートを Autoware 座標系で設定する

4-4-4. 外部連携インターフェース

① 【IF301】 Cesium、SENSYN CORE Pilot サーバー間

- インターフェースの概要
 - Cesium と SENSYN CORE Pilot 間で通信を行う
- 本インターフェースを利用する機能
 - 【FN014】 【FN016】
- プロトコル
 - HTTPS

表 4-30 Cesium、SENSYN CORE Pilot サーバー間インターフェース

インターフェース名	内容
地図データリクエスト	Cesium から地図データを読み込む
3D オブジェクトデータリクエスト	Cesium から 3D オブジェクトのデータを読み込む

② 【IF302】 PLATEAU 配信サービス、SENSYN CORE Pilot サーバー間

- インタフェースの概要
 - PLATEAU 配信サービスと SENSYN CORE Pilot 間で通信を行う
- 本インタフェースを利用する機能
 - 【FN014】 【FN016】 【FN019】
- プロトコル
 - HTTPS
- その他
 - PLATEAU 配信サービスについて、詳しくは GitHub ドキュメントを参照⁷

表 4-3 1 PLATEAU 配信サービス、SENSYN CORE Pilot サーバー間インタフェース

インタフェース名	内容
地形データリクエスト	PLATEAU 配信サービスから PLATEAU-Terrain データを読み込む
建築物 LOD2 データのリクエスト	PLATEAU 配信サービスから建築物 LOD2 データを読み込む

③ 【IF303】 場所を特定できる文字列を緯度・経度座標に変換するためのジオコードデータベース

- インタフェースの概要
 - Cesium Geocode API
 - 場所を特定できる文字列を緯度・経度座標に変換するためのジオコードデータベース
- 本インタフェースを利用する機能
 - 【FN014】
- プロトコル
 - HTTPS
- その他
 - 詳しくは Cesium 公式ドキュメントを参照⁸

表 4-3 2 場所を特定できる文字列を緯度・経度座標に変換するためのジオコードデータベース

Name	Type
container	Element string
Scene	Scene
geocoderServices	Array
autoComplete	boolean
flightDuration	number
destinationFound	Geocoder.DestinationFoundFunction

⁷<https://github.com/Project-PLATEAU/plateau-streaming-tutorial>⁸<https://cesium.com/learn/cesiumjs/ref-doc/Geocoder.html>

④ 【IF304】 ELAS、DJI ドローン間

- インタフェースの概要
 - ELAS と DJI ドローン間で通信を行う
- 本インタフェースを利用する機能
 - 【FN003】
- プロトコル
 - シリアル通信 (RS232)

表 4-3 3 ELAS、DJI ドローン間インタフェース

インタフェース名	内容
RTK-GNSS 測位情報リクエスト	RTK-GNSS によるドローン自己位置測位情報をリクエスト
FPV 撮影画像リクエスト	ドローン内蔵の前方カメラ (FPV) で撮影した画像をリクエスト
ドローン制御リクエスト	ドローンの制御をリクエスト

4-5. 実証に用いたデータ

4-5-1. 活用したデータ一覧

1) 利用した 3D 都市モデル

- 年度：2024 年度（西暦）
- 都市名：大阪府大阪市
- ファイル名：27100_osaka-shi_city_2024_citygml_1_op
- メッシュ番号：51357380（インデックスマップで黄色囲いの箇所）

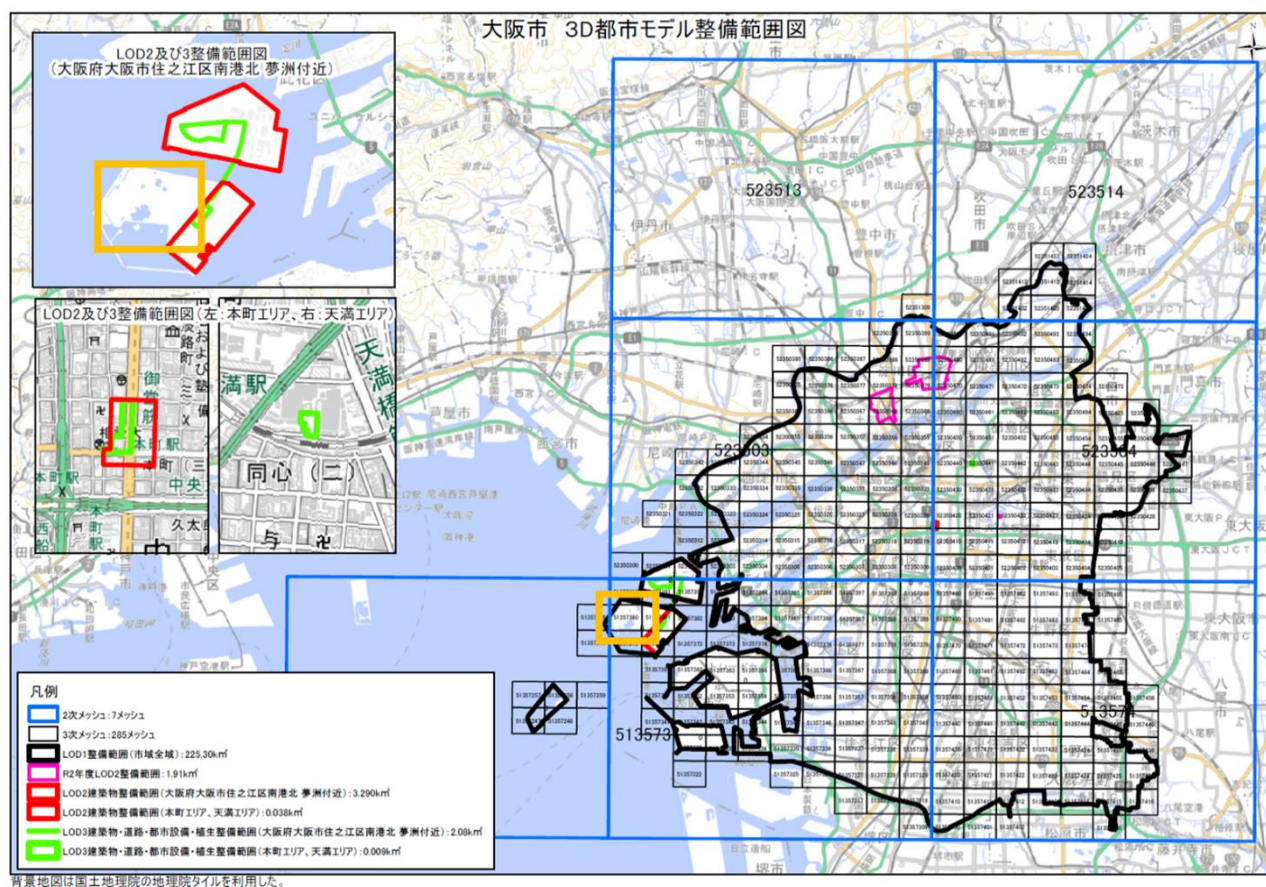


図 4-64 インデックスマップ（大阪市）

表 4-3 4 利用する 3D 都市モデル

地物	地物型	属性区分	ID	属性名	内容	データを利用した機能 (ID)
建築物 LOD3	bldg:Building	空間属性	DT001	bldg:lod3Solid	LOD3 を構成する面の幾何形状	FN001
建築物 LOD4	bldg:Building	空間属性	DT002	bldg:lod4Solid	LOD4 を構成する面の幾何形状	FN001
道路 LOD3	tran:Road	空間属性	DT003	tran:lod3MultiSurface	道路縁により囲まれた道路の範囲	FN101
都市設備 LOD3	frn:CityFurniture	空間属性	DT004	frn:lod3Geometry	LOD3 を構成する面の幾何形状	FN001 FN101
地形	Geo:GeographicalFeatures	空間属性	DT005	Geo:GeographicalFeatures	地形モデル	FN101

2) 利用するその他のデータ

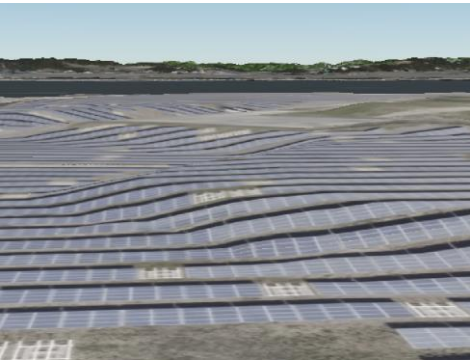
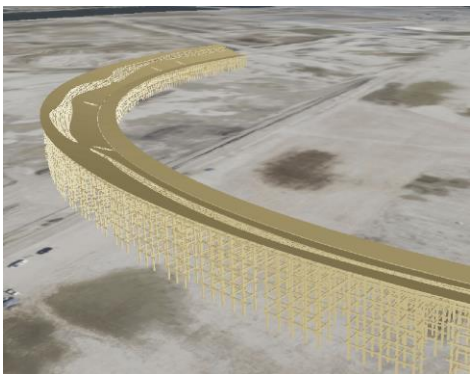
1. データ一覧

表 4-3 5 利用するその他データ (一覧)

ID	エリア (都市)	活用データ	内容	データ形式	更新情報	出所	データを利用した機能 (ID)
DT101	大阪市	BIM (IFC)	大阪・関西万博パビリオンの設計図面を基に、今回の実証実験のために新規作成した BIM モデル	IFC	2024 年度	大阪市、株式会社竹中工務店	FN001
DT102	大阪市	PLATEAU-Terrain	地形モデルの Terraindb データ	Terraindb	2023 年度	PLATEAU 配信サービス	FN014
DT103	大阪市	PLATEAU-3DTiles	建築物モデル等の 3D Tiles データ	3D Tiles	2024 年度	国際航業株式会社	FN014

2. データサンプル（イメージ）

表 4-3 6 利用するその他データ（サンプル）

ID	活用データ	サンプル・イメージ	
DT101	BIM（IFC）		
DT102	PLATEAU-Terrain		
DT103	PLATEAU-3DTiles		

4-5-2. 生成・変換したデータ

表 4-3 7 生成・変換するデータ

ID	システムに入力するデータ （データ形式）	用途	処理内容	データ処理ソフトウェア	活用データ （データ形式）	データを利用した機能（ID）
----	-------------------------	----	------	-------------	------------------	----------------

DT201	3D 都市モデル (CityGML 形式)	点群マッチングのベースとなる点群マップを生成する	● 3D 都市モデル (CityGML 形式) を PLATEAU PCL Generator に読み込み、点群マップを生成する	PLATEAU PCL Generator	PCD 形式	FN101
DT202	LiDAR スキャンデータ (ROS メッセージ形式)	自己位置推定のため	● 3D LiDAR センサーで計測 ● 【FN004】 LiDAR ローカリゼーション機能で自己位置推定処理を行う	【FN004】 LiDAR ローカリゼーション	Point Cloud	FN004 FN011
DT203	GNSS 測位座標 (NMEA 形式)	自己位置測位のため	● GNSS を利用して、ドローンの位置座標を取得 ● 【FN007】 GNSS 測位情報チェッカー機能、【FN008】 GNSS 測位情報座標変換機能でそれぞれ自己位置推定に必要な処理を行う	【FN007】 GNSS 測位情報チェッカー機能 【FN008】 GNSS 測位情報座標変換機能	GNSS 測位座標 (NMEA 形式)	FN007 FN008 FN010
DT204	IMU (機体) 加速度/角速度 (ROS メッセージ形式)	自己位置推定のため	● ジャイロと加速度計によって 3 次元の加速度/角速度を計測 ● 【FN010】 測位情報統合機能で、自己位置推定のために利用	【FN010】 測位情報統合機能	加速度/角速度 (ROS メッセージ形式)	FN010
DT205	カメラ画像データ (ROS メッセージ形式)	自己位置推定のため	● ドローン内蔵カメラによって画像を記録 ● 【FN009】 ビジュアルイナーシャルオドメトリ (VIO) 機能で、自己位置推定のために利用	【FN009】 ビジュアルイナーシャルオドメトリ (VIO) 機能	画像データ (Image 形式)	FN009

4-6. ユーザーインターフェース

4-6-1. 画面一覧

1) PC 用画面

表 4-38 PC 画面一覧

ID	連携 (ID)	画面名	画面説明	関連する機能 (ID)
SC001	-	フライトプラン	自動飛行を実行するエリアの名称を規定し、新規ルートの作成やインポートした CityGML から生成したオブジェクトモデルをひも付けるための登録画面	FN015
SC002	-	新規ルートの作成	フライトプラン登録後呼び出したエリアにおいて、新規の運航ルートを作成する際に必要な情報・運航方法を決定する画面	FN015
SC003	SC004,005,006,007	メイン画面	モデルの地図表示、車両・ドローンの選択、自動運航ルートの作成と保存までを一元表示	FN015
SC004	-	機体・カメラ選択	リスト形式で、今回利用する車両・ドローン及びカメラを選択	FN015
SC005	SC006	アクション表示 (車両)	- 車両のアクションを規定、ウェイポイント、速度の設定を実施 - 速度はスライダーバー又は数値を直接入力 - 走行ルートを保存	FN015
SC006	SC005,007	地図画面	3D 都市モデルを表示 3D 形式のマップで、任意の方向、アングルの変更、縮尺の変更が可能 - ウェイポイントの設定が可能 - 作成したルートを地図上にリアルタイム反映	FN015
SC007	SC006	アクション表示 (ドローン)	- ドローンのアクションを規定、ウェイポイント、高度、速度の設定を行う - 高度、速度はスライダーバー又は数値を直接入力 - 飛行ルートを保存	FN015
SC008	SC009	ルート統合画面	複数のルートを統合して、車両とドローンが連続運航するルートを作成	FN015
SC009		ルート選択	連続自動運航のために統合する車両・ドローン各ルートを選択	

SC010	-	ルートアップロード	● 車両・ドローンにルートを送り、走行開始画面に遷移	FN015
SC011	-	走行開始画面	● 車両の走行開始時に利用	FN015
SC012	SC013,014,015,016	モニタリング画面	● 運航中の車両・ドローンの位置を地図画面に表示 ● サブウィンドウにテレメトリーとして、現在位置座標、高度、速度、LiDAR 測位状態、バッテリー情報等の表示 ● 緊急時の停止、再開、帰還の選択	FN019
SC013	SC014	車両テレメトリー画面	● 現在位置座標、速度等を表示	FN019
SC014	SC015	走行完了・飛行開始画面	● 車両の走行完了後に表示され、ドローンの飛行開始時に利用	FN019
SC015	SC016	ドローンテレメトリー画面	● 現在位置座標、高度、速度、GNSSRTK 測位状態、LiDAR 測位状態、バッテリー情報等を表示	FN019
SC016	-	緊急停止	● ドローン飛行の一時停止実行	FN020

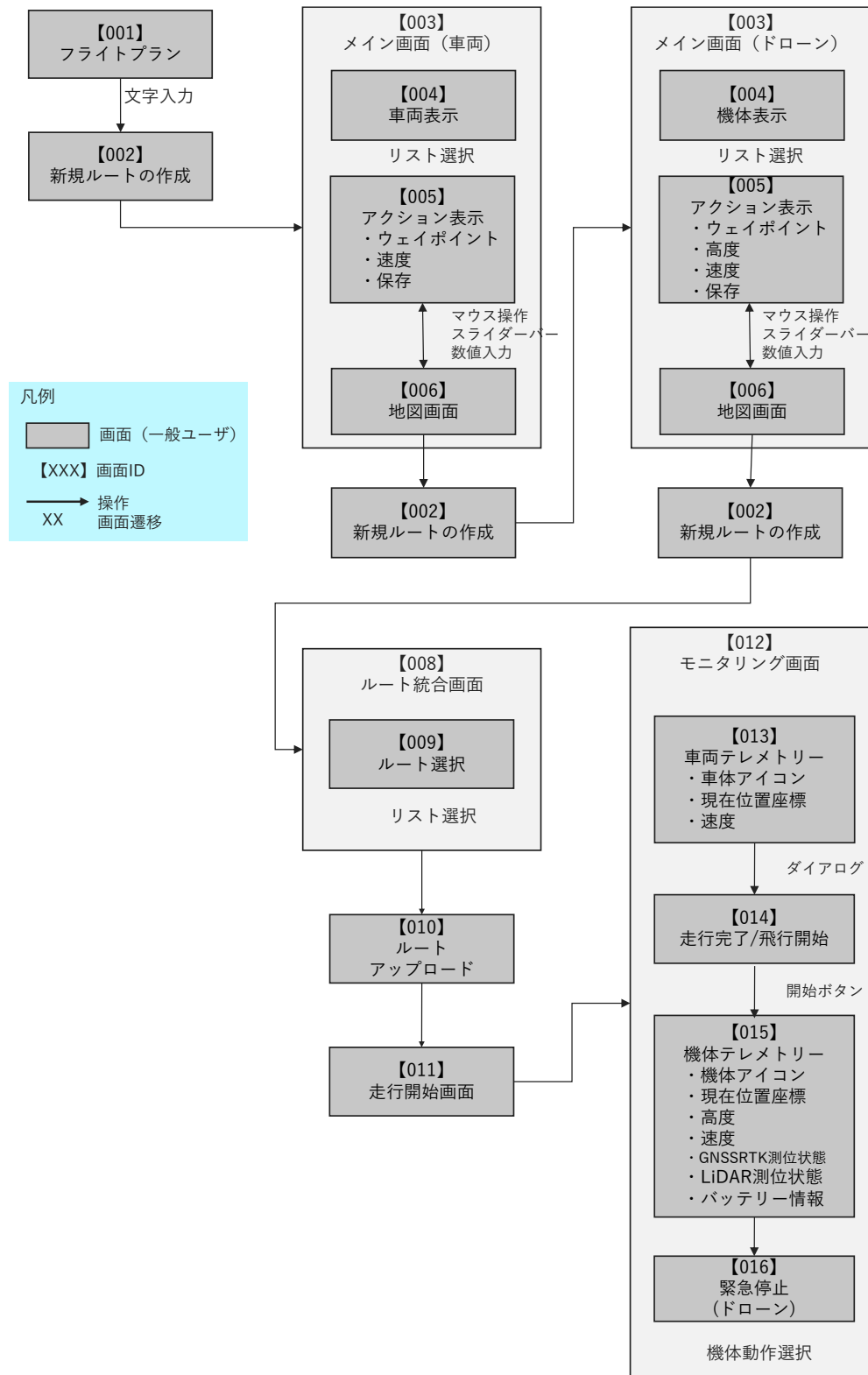
2) 管理者用画面

表 4-39 管理者用画面一覧

ID	連携 (ID)	画面名	画面説明	関連する機能 (ID)
SC101	-	動作選択	● メイン画面の3点リーダーから、リスト選択で新規 3D オブジェクトの読み込みを選択	FN015
SC102	-	新規 3D オブジェクトの登録	● CityGML や BIM から変換したオブジェクトを SENSYN CORE Pilot へインポートする ● ID 入力で自動的に反映	FN015
SC103	-	3D オブジェクトの表示	● 地図画面にインポートしたオブジェクトを表示	FN015

4-6-2. 画面遷移図

1) PC 用画面



2) 管理者用画面

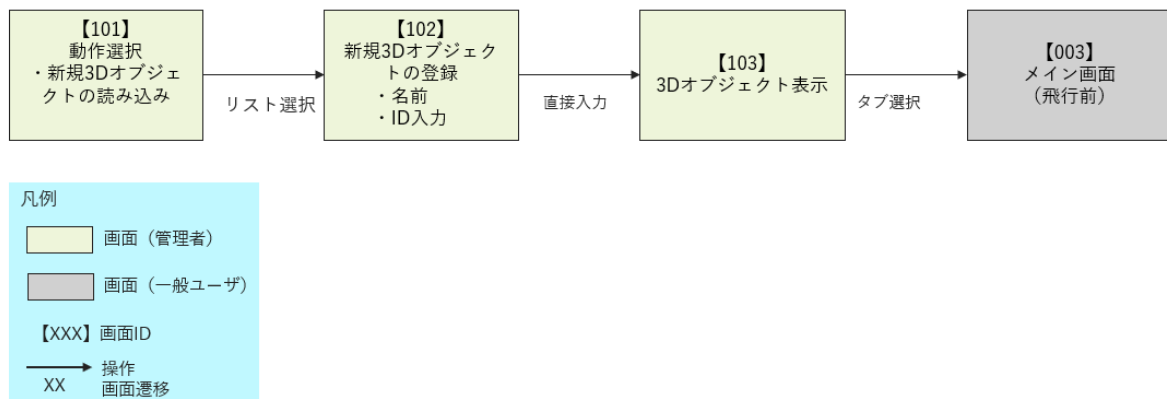


図 4-66 管理者用画面遷移図

4-6-3. 各画面仕様詳細

1. 【SC001】フライトプラン

- 概要
 - 事前にインポートした 3D モデルがマージされているエリアを選択
 - フライトプランを命名し、配下に新規ルートを複数登録することを可能にする
- 画面イメージ

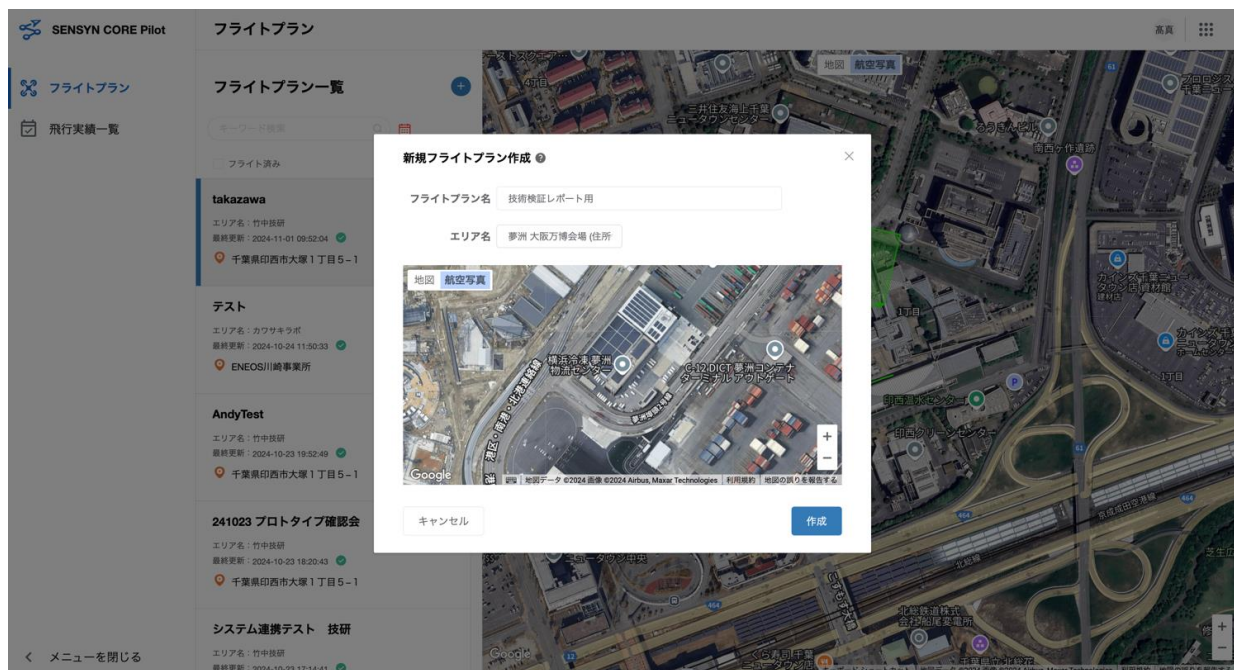


図 4-67 フライトプランの作成画面

2. 【SC002】新規ルートの作成

● 概要

- フライトプランで設定された対象エリアにおけるルート作成の初期画面
 - ルート名を入力
 - 運航方法の選択
- ※今回の検証車両・ドローンはウェイポイント運航のみ対応

● 画面イメージ

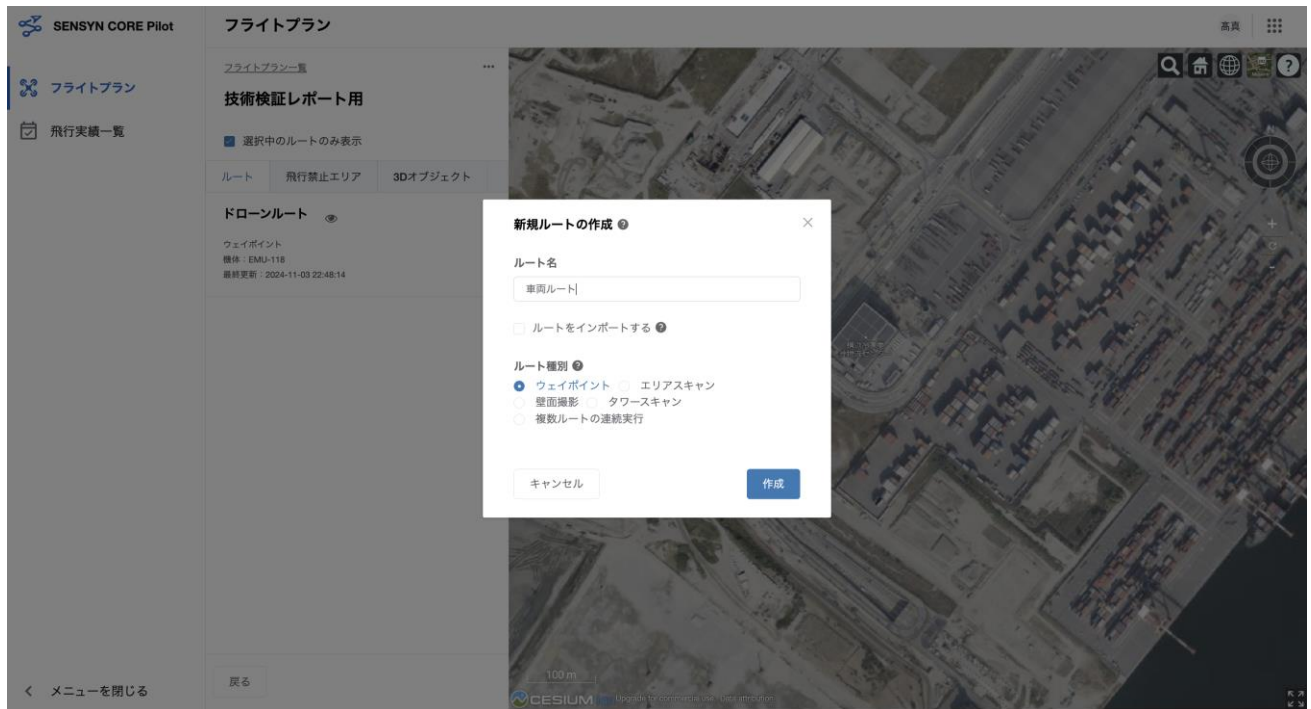


図 4-68 新規ルートの作成画面

3. 【SC003】メイン画面（車両・ドローン）

- 概要

- 地図画面と車両・ドローンの選択、アクションの入力画面に大きく分けられ、ユーザーは、地図を見ながら走行ルートを作成
- 編集ボタンから、走行ルートの修正やアクションの追加が可能

- 画面イメージ

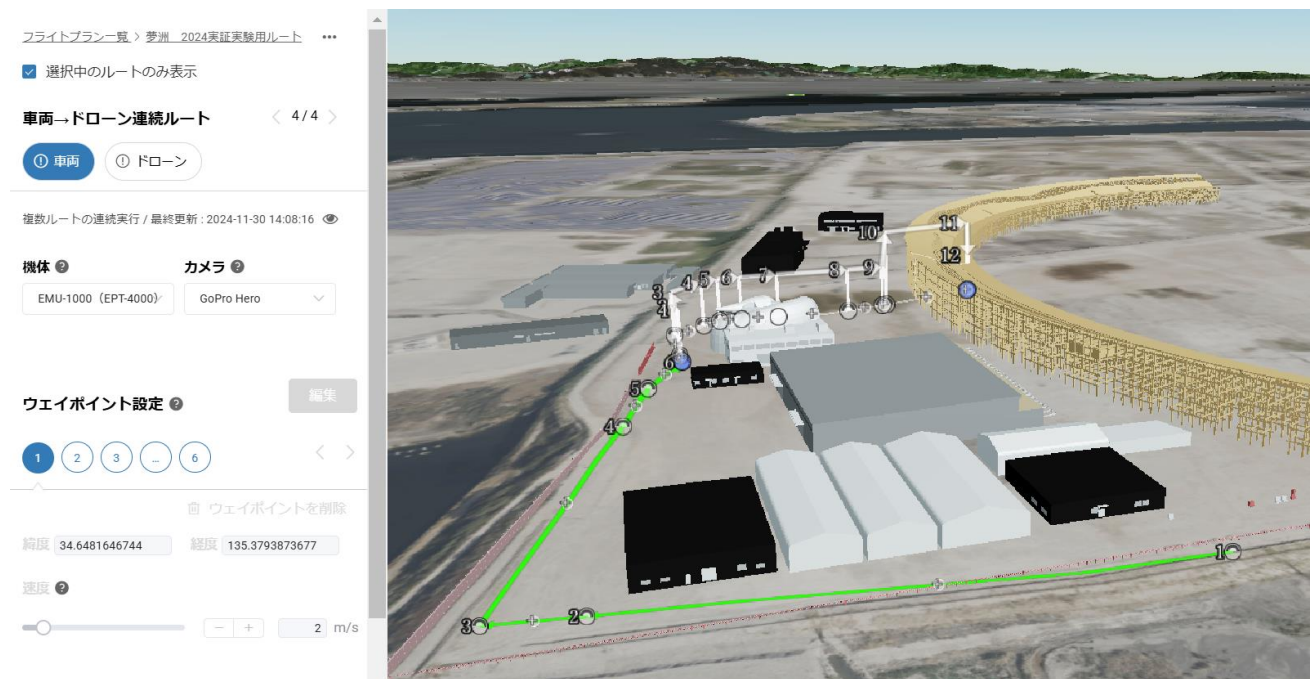


図 4-69 メイン画面のイメージ

4. 【SC004】機体・カメラ選択

- 概要
 - 当該ルートで使用する車両又はドローンを選択
※車両・ドローンの事前登録は管理者が実施
 - 使用するカメラを選択
- 画面イメージ



図 4-70 機体選択画面イメージ

5. 【SC005】アクション表示(車両)

- 概要
 - 車両のアクションを規定、ウェイポイント、速度の設定を実施
 - 速度はスライダーバー又は数値を直接入力
 - 走行ルートを保存
- 画面イメージ



図 4-71 アクション表示画面(車両)イメージ

6. 【SC006】 地図画面 ※車両・ドローン共通

- 概要
 - 3D 都市モデルを表示
 - 3D 形式のマップで、任意の方向、アングルの変更、縮尺の変更が可能
 - ウェイポイントの設定が可能
 - 作成したルートを実時に地図上にリアルタイム反映
- 画面イメージ

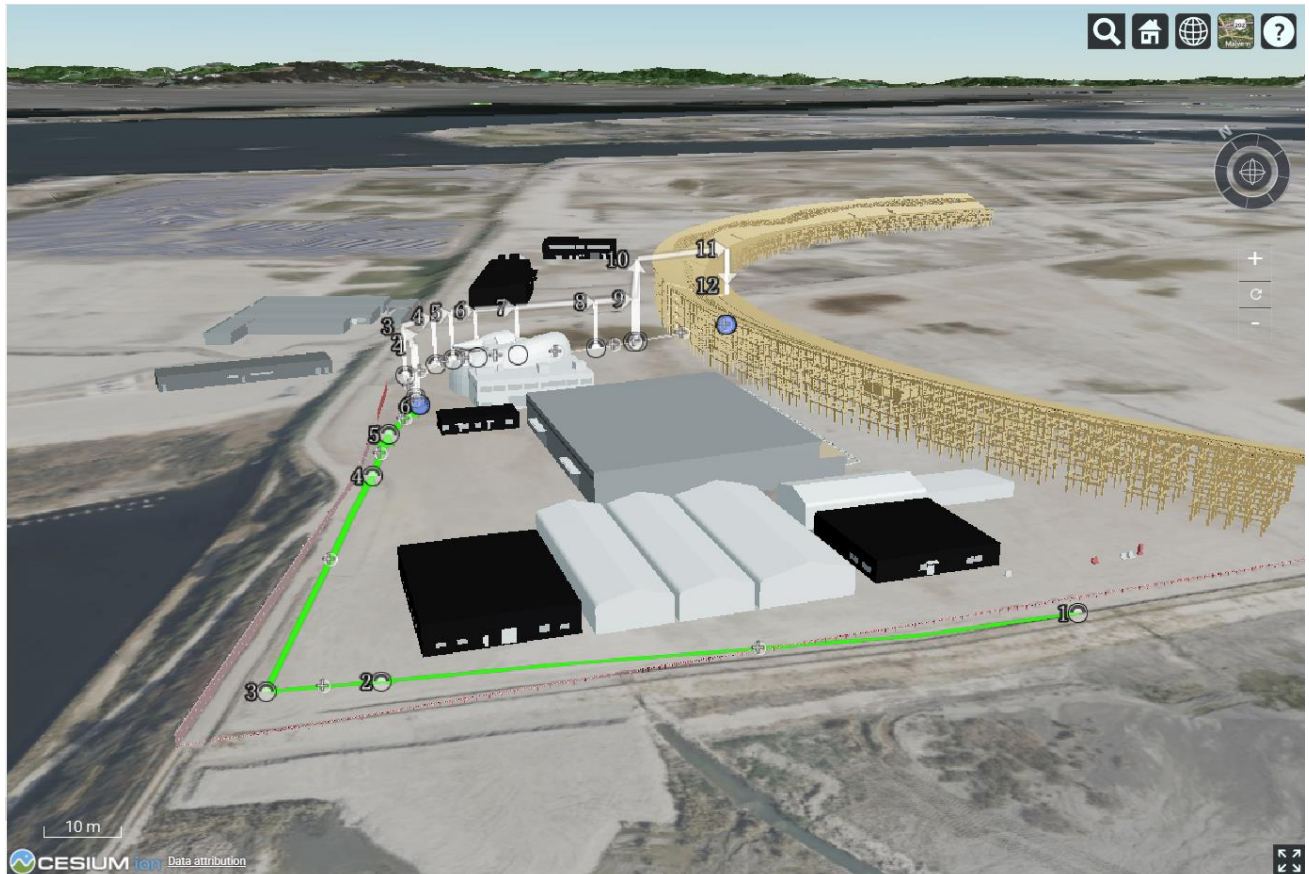


図 4-72 地図画面イメージ

7. 【SC007】アクション表示(ドローン)

● 概要

- 緯度経度を直接入力して、ウェイポイントを設定
※地図画面でポインタ操作により任意の地点でダブルクリックでも可能
- ドローンの飛行速度を設定
※スライダーバー又は数値の入力
- ドローンの飛行高度を設定
※スライダーバー又は数値の入力
- ルートの保存

● 画面イメージ

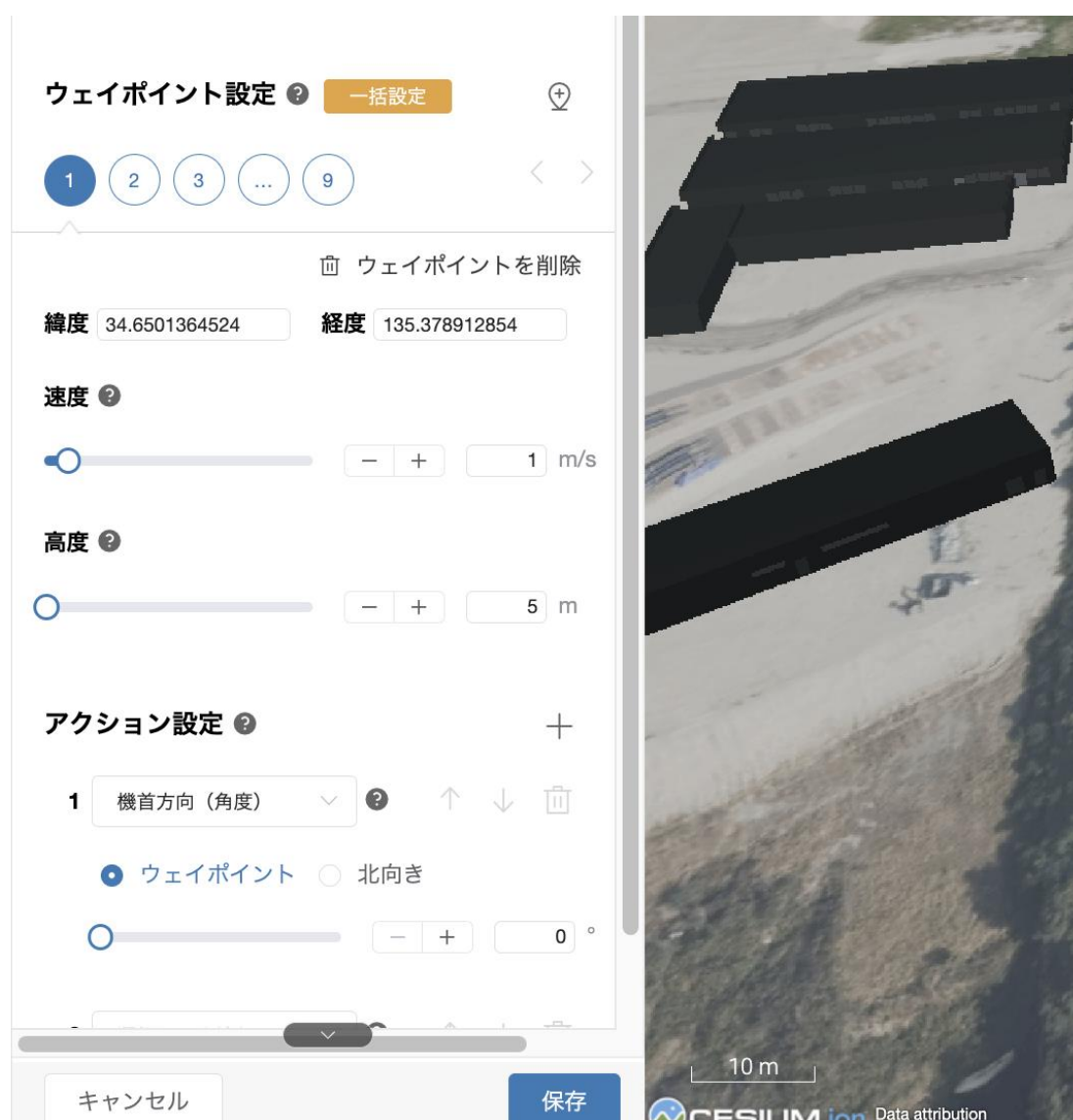


図 4-73 アクション表示画面(ドローン)イメージ

8. 【SC008】ルート統合画面

- 概要
 - 複数のルートを統合して、車両とドローンが連続運転するルートを作成する
- 画面イメージ

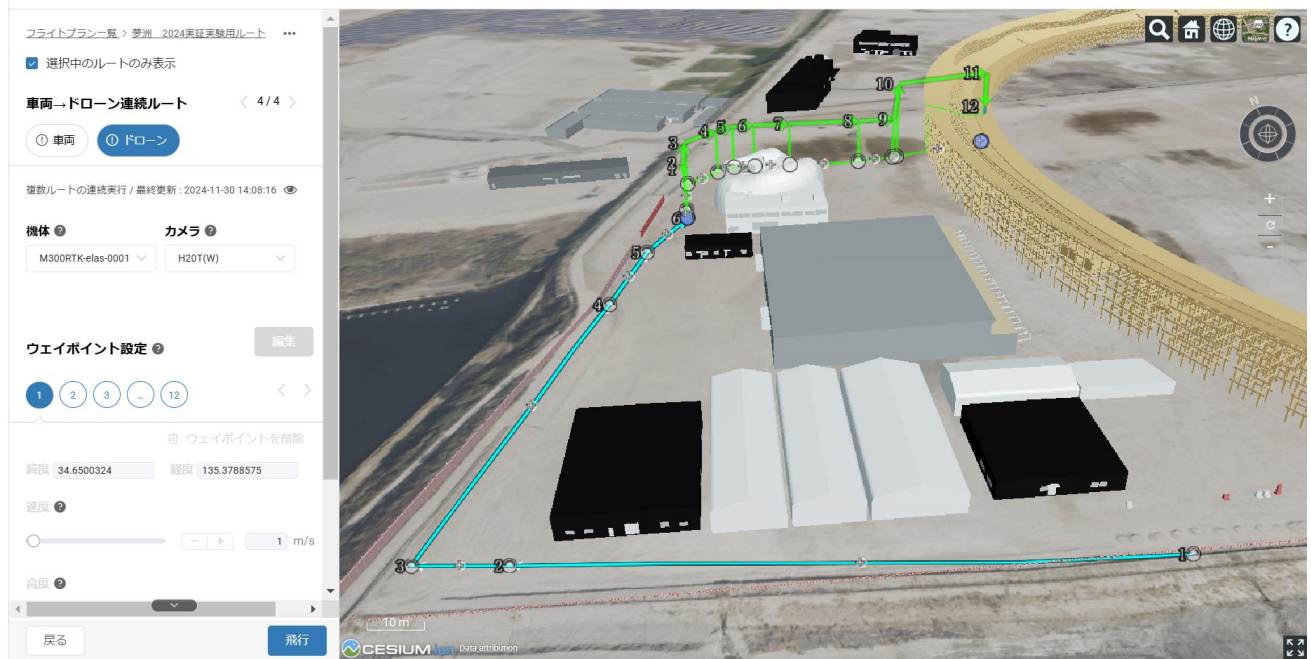


図 4-74 ルート統合画面イメージ

9. 【SC009】ルート選択

- 概要
 - 連続自動運転のために統合する車両・ドローン各ルートを選択
- 画面イメージ

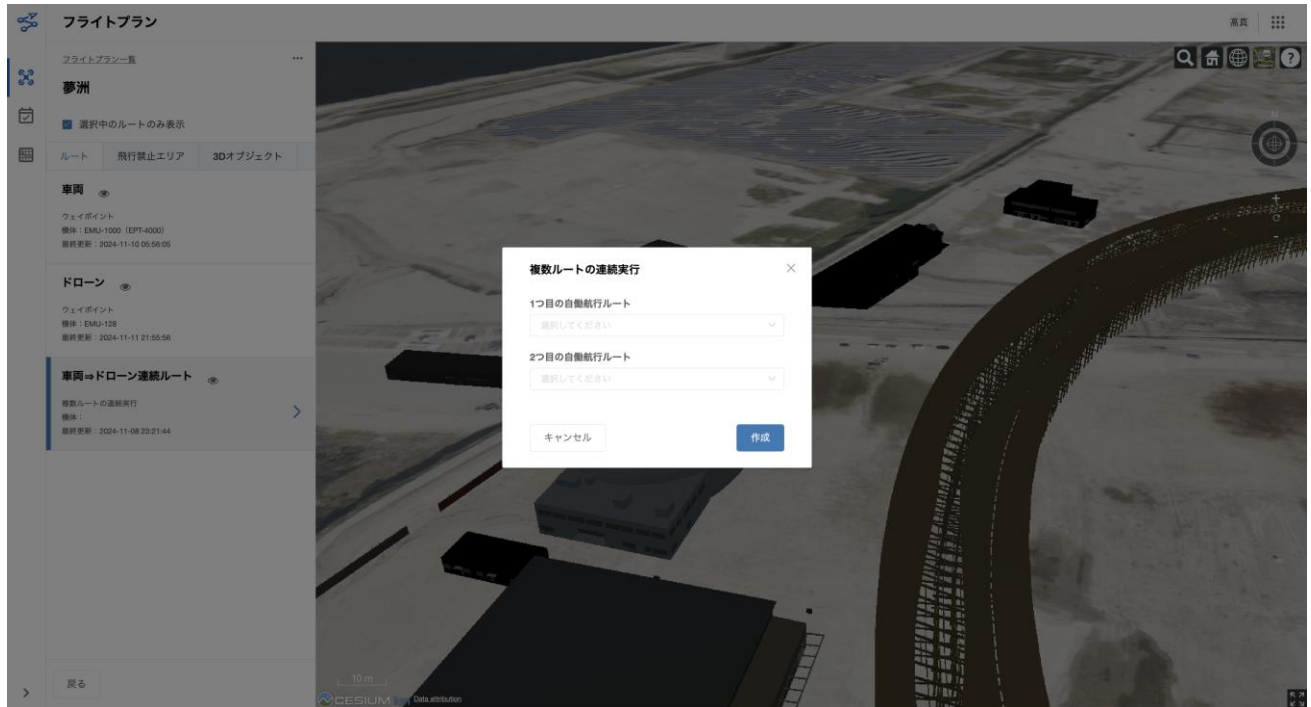


図 4-75 ルート選択イメージ

10. 【SC010】ルートアップロード

- 概要
 - 車両・ドローンにルートを転送し、走行開始画面に移移
- 画面イメージ

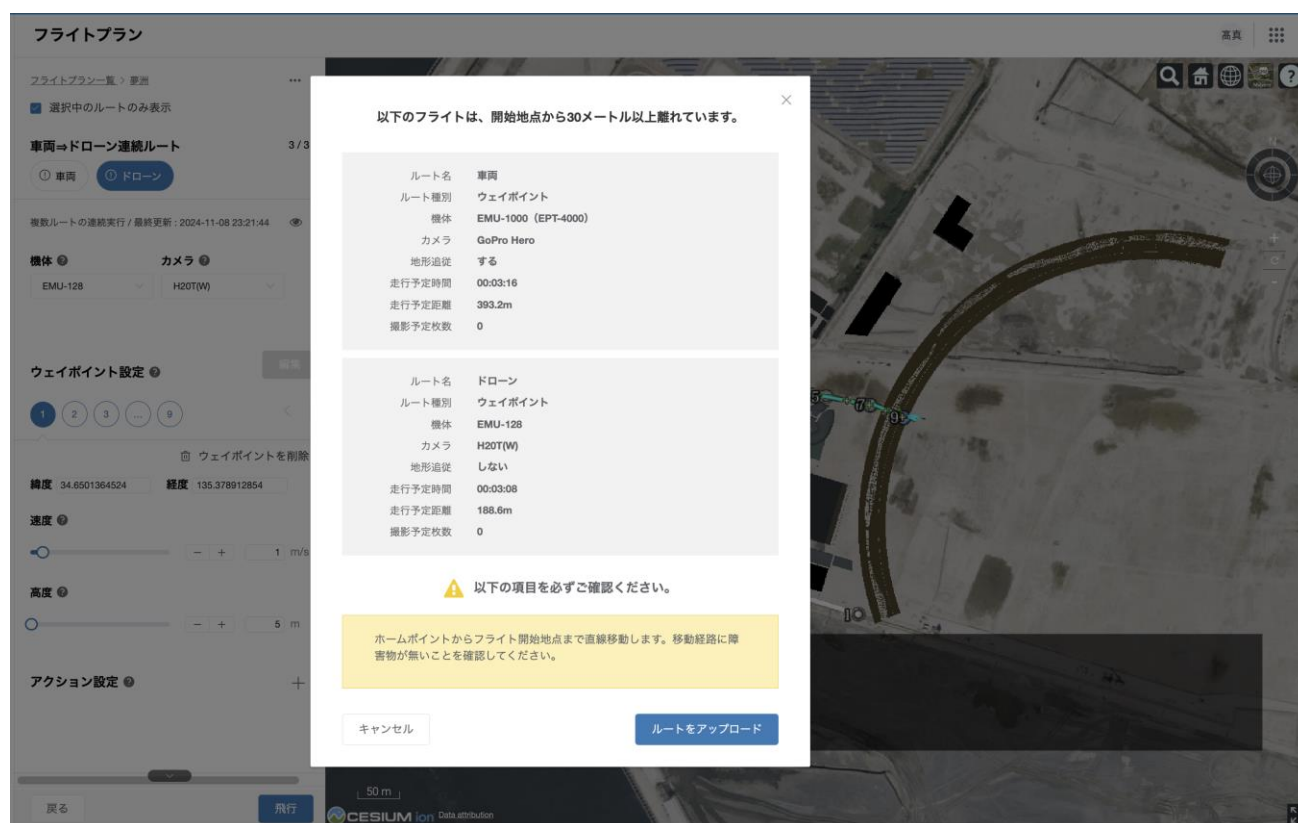


図 4-76 ルートアップロード画面イメージ

11. 【SC011】 走行開始画面

- 概要

- ルートアップロード後にすぐに自動で走行しないために安全の観点で設ける
- ポップアップ画面には、走行前の注意事項を改めて提起
- 周辺状況と機体の最終チェック後、走行開始ボタンを選択すると、車両が自動で走行

- 画面イメージ

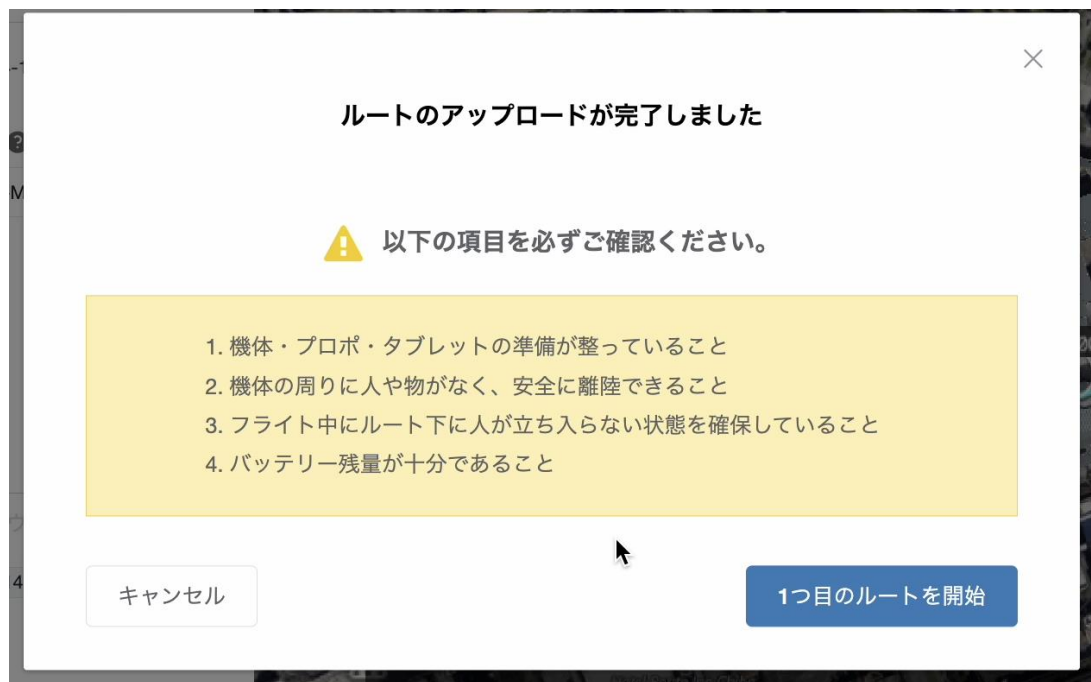


図 4-77 走行開始画面イメージ

12. 【SC012】モニタリング画面

- 概要

- 自動運航を行っている際の監視画面、車両・ドローンの現在位置を3次元で地図画面に表示

- 画面イメージ

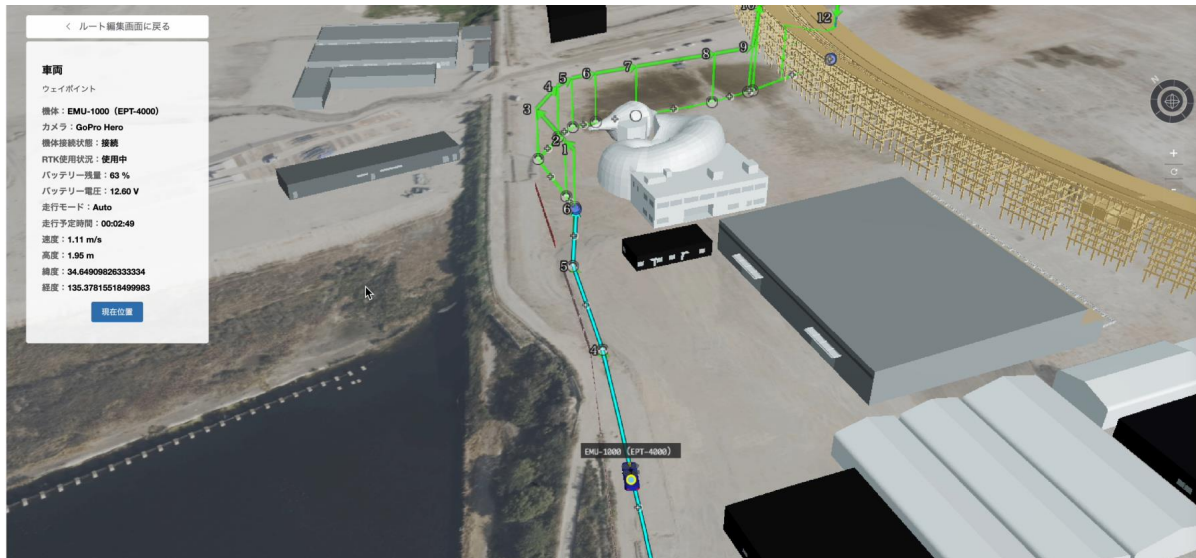


図 4-78 モニタリング画面イメージ

13. 【SC013】車両テレメトリー画面

- 概要
 - 車両の緯度経度座標、速度等を表示
- 画面イメージ



図 4-79 車両テレメトリーの画面イメージ

14. 【SC014】 走行完了・飛行開始画面

- 概要

- 車両の走行完了後に表示され、ドローンの飛行開始時に利用
- 車両の走行完了を確認するため、また車両の走行完了後にドローンが自動で飛行しない等、安全性の観点で設ける

- 画面イメージ

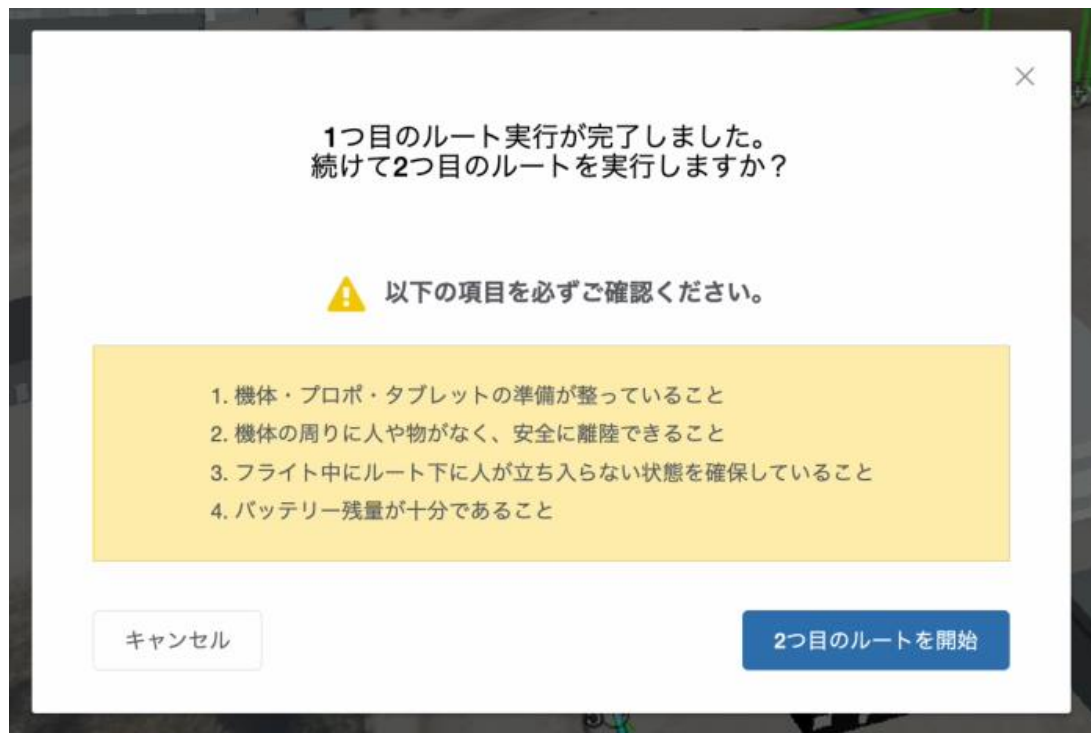


図 4-80 走行完了・飛行開始画面イメージ

15. 【SC015】ドローンテレメトリー画面

- 概要

- ドローンの緯度経度座標、高度、速度、GNSSRTK 測位状態、LiDAR 測位状態、バッテリー情報等を表示

- 画面イメージ

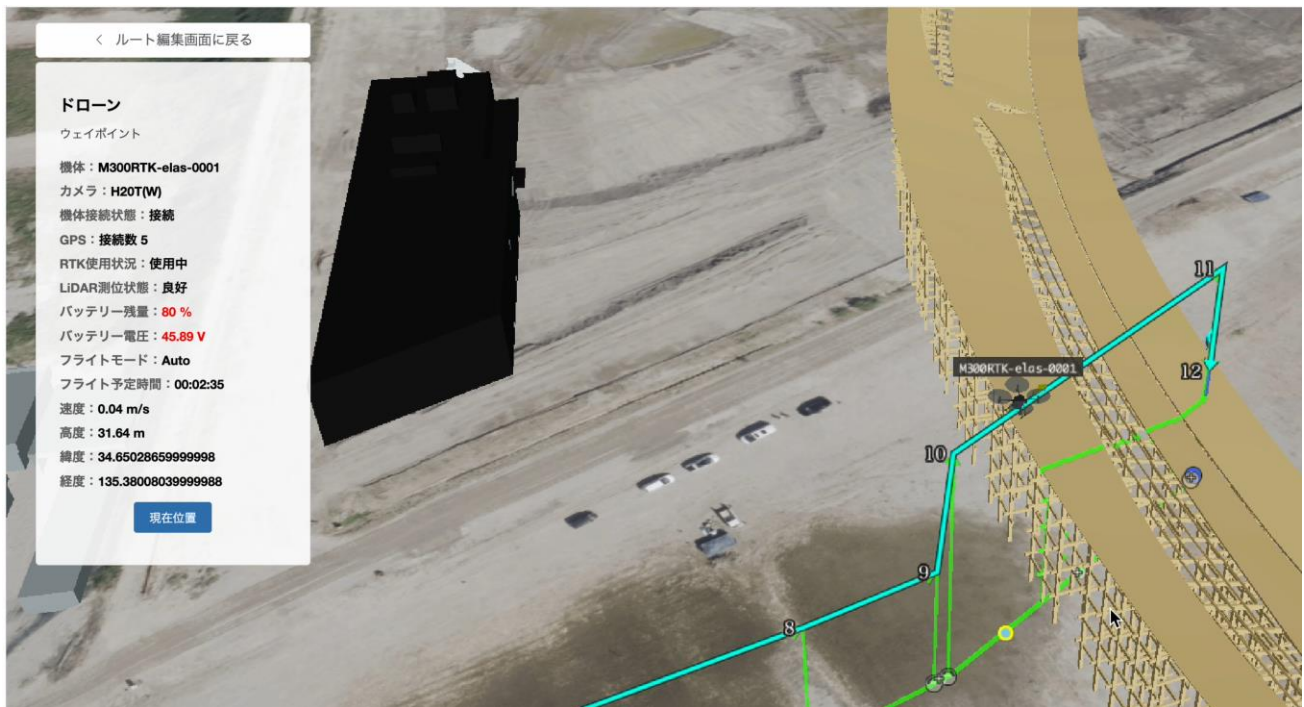


図 4-81 ドローンテレメトリーの画面イメージ

16. 【SC016】 緊急停止

- 概要
 - ドローンの飛行中に何らかの影響で機体を緊急停止させる際に利用
 - 機体の一時停止のほか、飛行再開と離陸地点への自動帰還の選択を具備
 - 【SC015】 ドローンテレメトリ画面右上の飛行機アイコンをクリックするとポップアップ
- 画面イメージ



図 4-82 緊急停止の画面イメージ

1) 管理者画面

1. 【SC101】 動作選択

- 概要

- メイン画面の3点リーダーから、リスト選択で新規 3D オブジェクトの読み込みを選択

- 画面イメージ

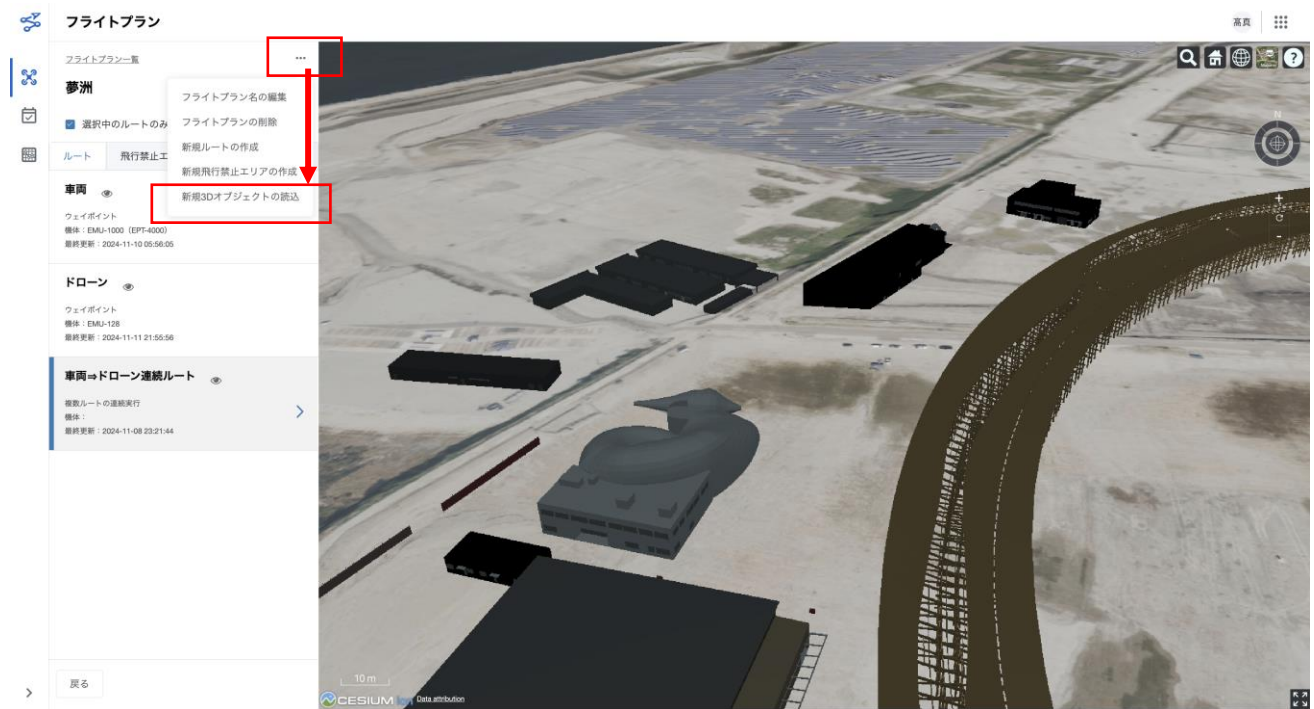


図 4-83 動作選択の画面イメージ

2. 【SC102】新規 3D オブジェクトの登録

- 概要
 - CityGML や BIM から変換したオブジェクトを SENSYN CORE Pilot へインポートする
 - ID 入力で自動的に反映
- 画面イメージ

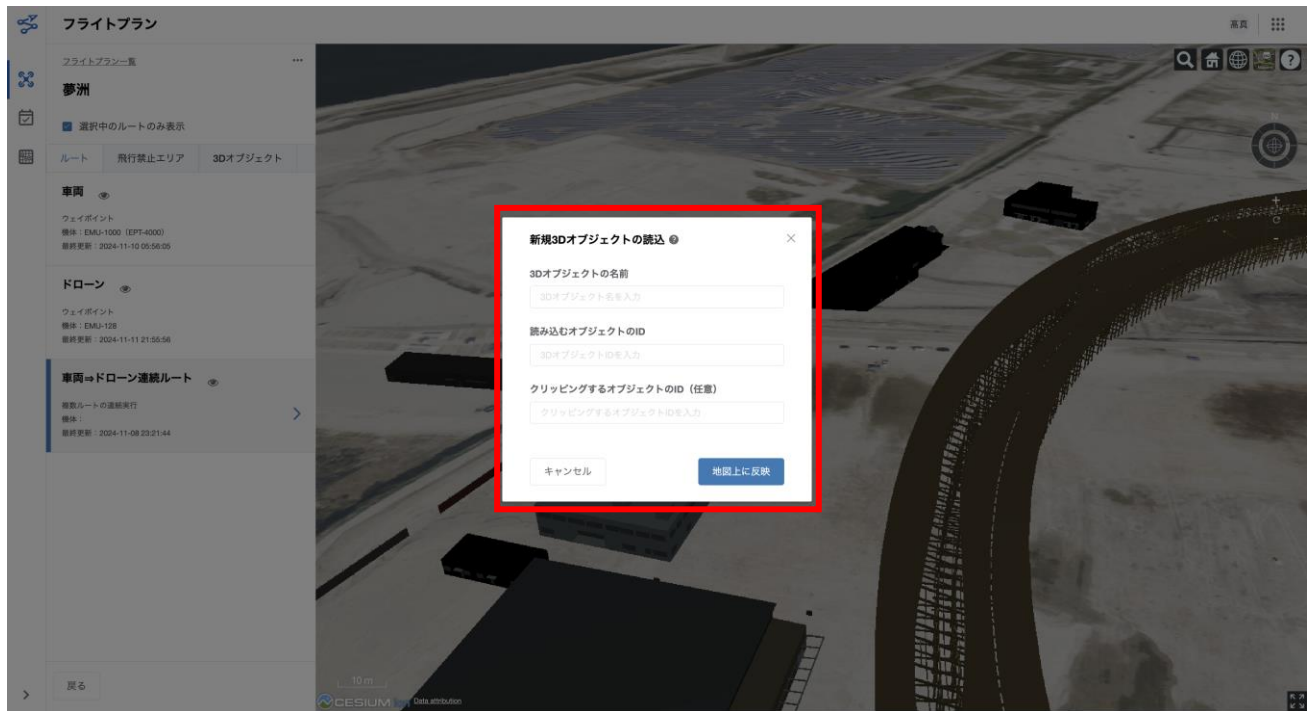


図 4-84 新規 3D オブジェクト読み込みの画面イメージ

3. 【SC103】 3D オブジェクトの表示

- 概要
 - 地図画面にインポートしたオブジェクトを表示
- 画面イメージ

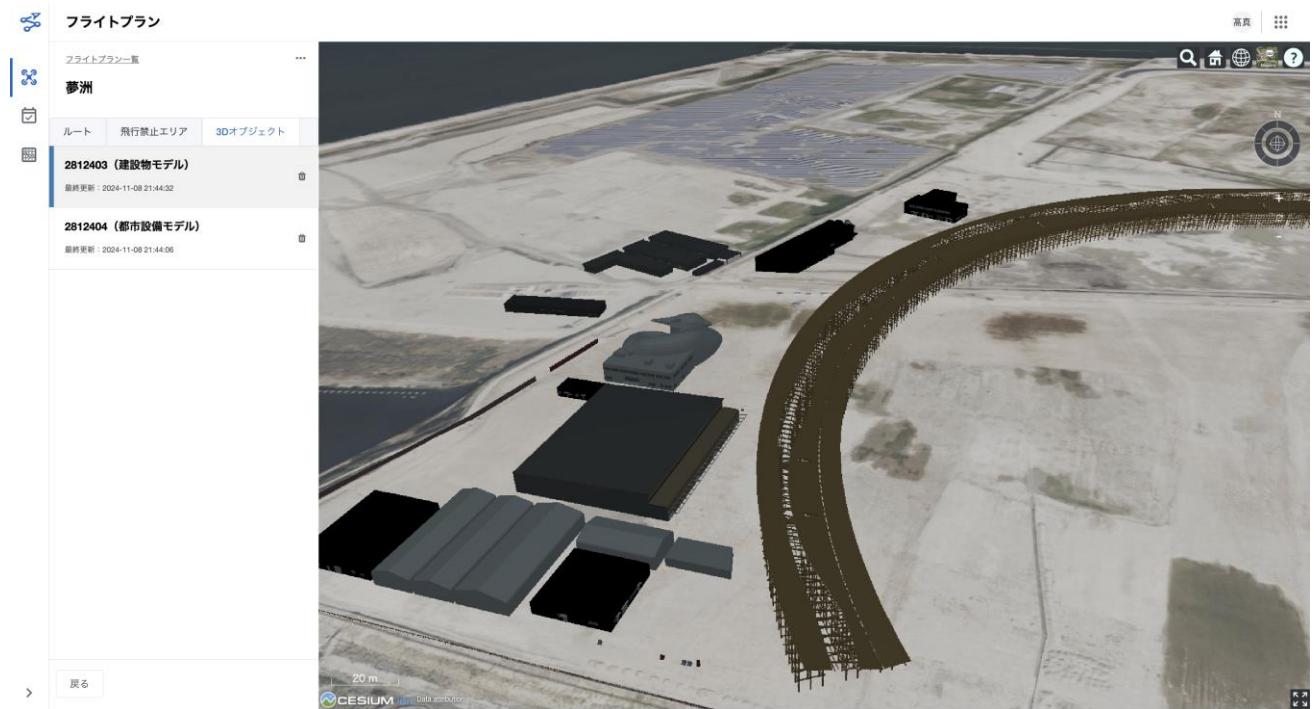


図 4-85 3D オブジェクトの表示画面イメージ

4-7. 実証システムの利用手順

4-7-1. 実証システムの利用フロー

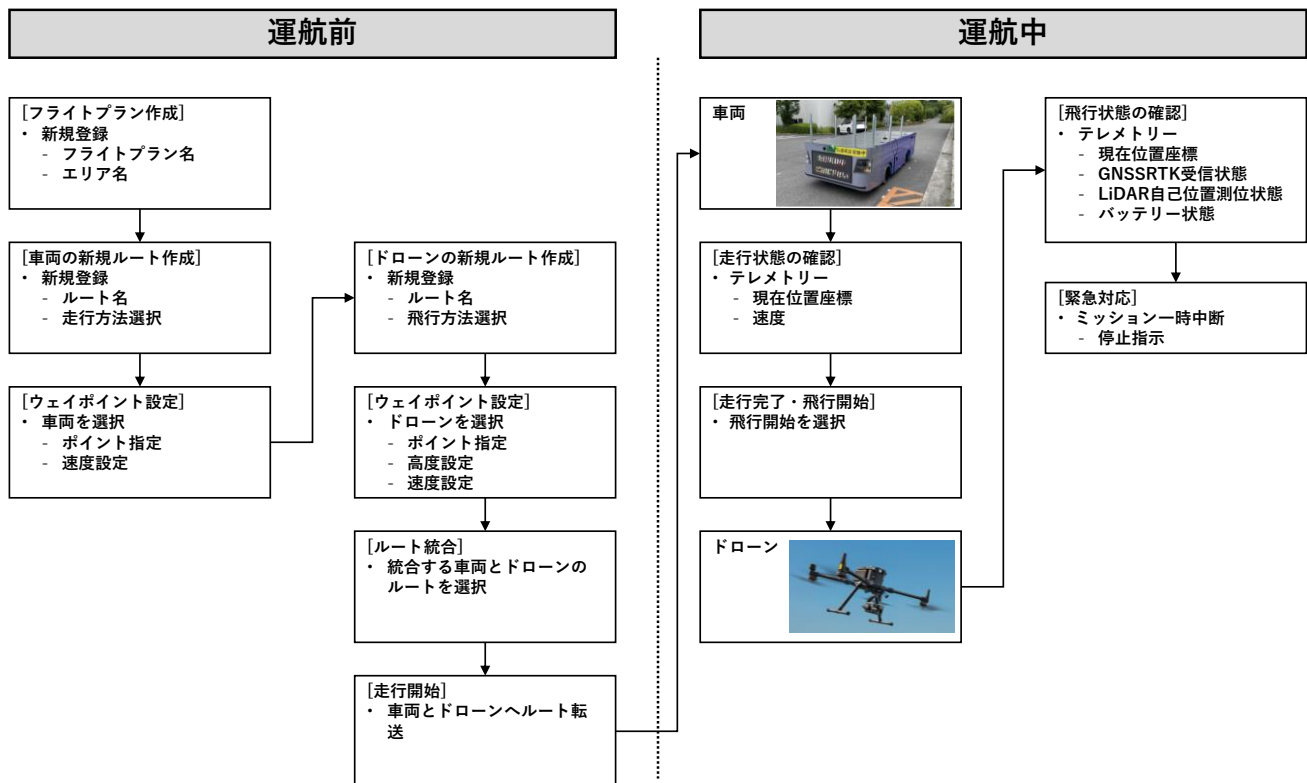


図 4-86 システムの利用フロー

- 最初にフライトプランを作成
- 車両の新規ルート作成時はルート命名して車両を選択、各ウェイポイントを指定後、速度を決定。ドローンの新規ルート作成時はルート命名してドローンを選択、各ウェイポイントを指定後、高度、速度を決定
- 車両・ドローンの連続運転のためルートを統合後、車両・ドローンへそれぞれのルートを転送
- 運転中は、車両・ドローンのテレメトリー情報をリアルタイムで取得
- 万一の場合は、ミッション一時中断して緊急停止

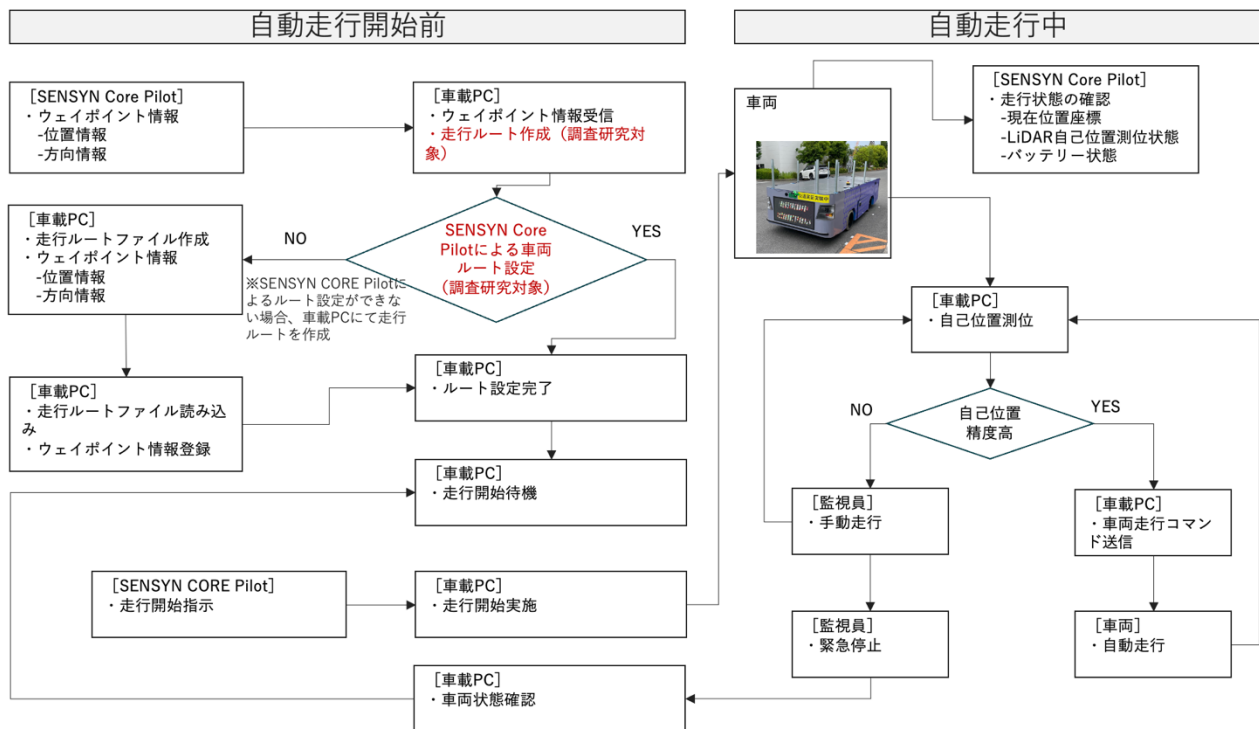


図 4-87 車両自動走行システムの利用フロー

- SENSYN CORE Pilot により、ルート作成、車載 PC によりルート受信（障害物が少ないルートを設定できるドローンに比べ、地上を走行する車両は、障害物に対する迂回設定や歩行者、対向車への走行に伴う操舵角や速度の加減速に関する車両特有の設定があり、SENSYN CORE Pilot から車両の制御を行うことが困難なため、今回の実証では SENSYN CORE Pilot からのルート情報を基にした走行ルート作成を調査研究対象とした。）
- 車載 PC で、走行ルート読み込み
- 車載 PC で、ルート設定完了。走行開始待機
- SENSYN CORE Pilot より、走行開始指示を受信。走行開始実施
- 走行中は SENSYN CORE Pilot で走行状態の確認
- 自己位置測位を行い、精度が高い場合は自動走行を実施
- 自己位置測位の精度が低い場合は手動で走行
- 緊急停止時は車両状態を確認し、走行開始待機状態となる

4-7-2. 各画面操作方法

1) PC 画面

1. 【SC001】フライトプラン

- 概要
 - 事前にインポートした 3D モデルがマージされているエリアを選択
 - フライトプランを命名、作成をクリック
- 画面イメージ
 - 「4-6-3.各画面仕様詳細」の記載と重複するため割愛

2. 【SC002】新規ルートの作成

- 概要
 - 「4-6-3.各画面仕様詳細」の記載と重複するため割愛
- 画面イメージ
 - 「4-6-3.各画面仕様詳細」の記載と重複するため割愛

3. 【SC003】メイン画面（車両・ドローン）

- 概要
 - 「4-6-3.各画面仕様詳細」の記載と重複するため割愛
- 画面イメージ
 - 「4-6-3.各画面仕様詳細」の記載と重複するため割愛

4. 【SC004】機体・カメラ選択

- 概要
 - 「4-6-3.各画面仕様詳細」の記載と重複するため割愛
- 画面イメージ
 - 「4-6-3.各画面仕様詳細」の記載と重複するため割愛

5. 【SC005】アクション表示(車両)

● 概要

- 緯度経度を直接入力して、ウェイポイントを設定
※地図画面でのポイント操作により、任意の地点でのダブルクリックでも設定可能
- 車両の走行速度を設定
※スライダーバー又は数値の入力
- アクション設定
- ルートの保存

● 画面イメージ



図 4-88 アクション表示画面(車両)イメージ

6. 【SC006】地図画面 ※車両・ドローン共通

● 概要

- 表示した 3D 都市モデルを参照し、任意の方向、アングルの変更、縮尺の変更を行いながらルートの確認、ウェイポイント設定を実施
- ウェイポイント設定時は、ポインタ操作のダブルクリックで新規ウェイポイントの追加が可能
- 作成したルートを地図上にリアルタイム反映

● 画面イメージ

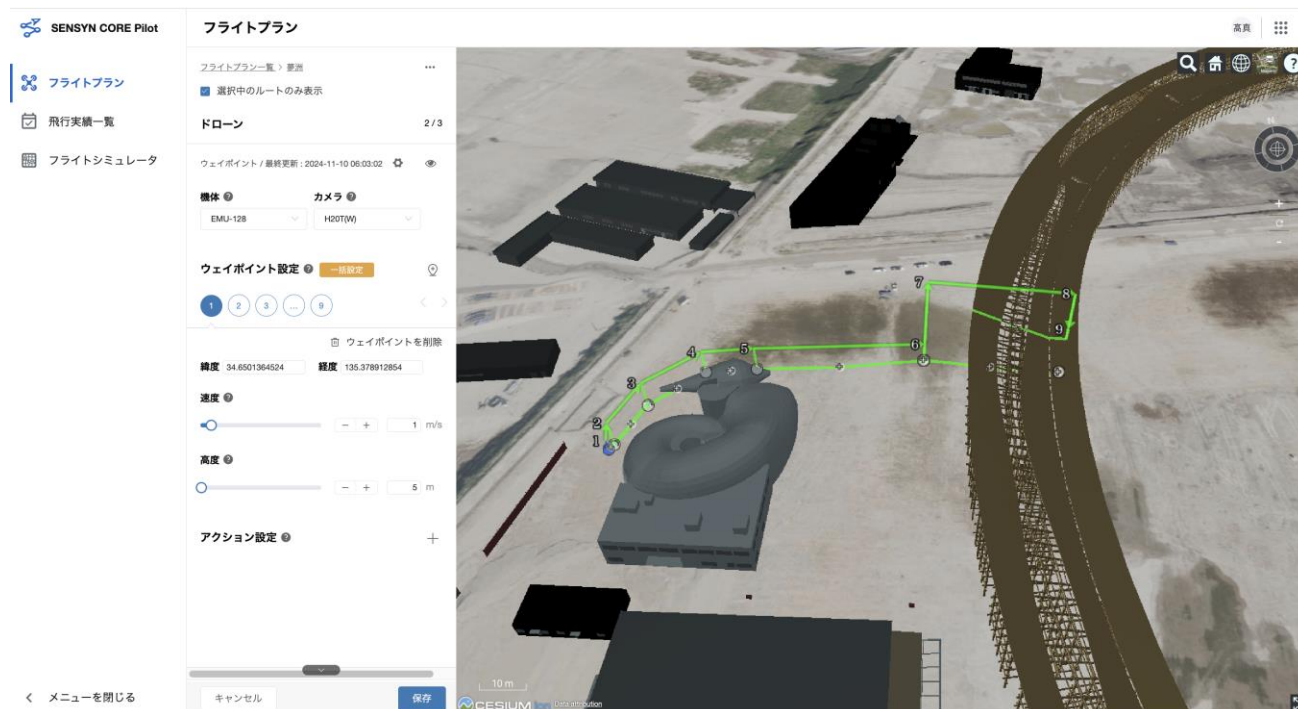


図 4-89 地図画面

7. 【SC007】アクション表示(ドローン)

● 概要

- 「4-6-3.各画面仕様詳細」の記載と重複するため割愛

● 画面イメージ

- 「4-6-3.各画面仕様詳細」の記載と重複するため割愛

8. 【SC008】ルート統合画面

- 概要
 - 複数のルートを統合し、車両とドローンが連続運航するルートを作成
 - 事前に、【SC005】車両ルート、【SC007】ドローンルートをそれぞれ用意
 - 【SC009】ルート選択画面で各ルートを選択、ルートを統合
- 画面イメージ
 - 「4-6-3.各画面仕様詳細」の記載と重複するため割愛

9. 【SC009】ルート選択

- 概要
 - 1 番目の自動運航として、車両のルートを選択
 - 2 番目の自動運航として、ドローンのルートを選択
 - ルート選択を決定
- 画面イメージ

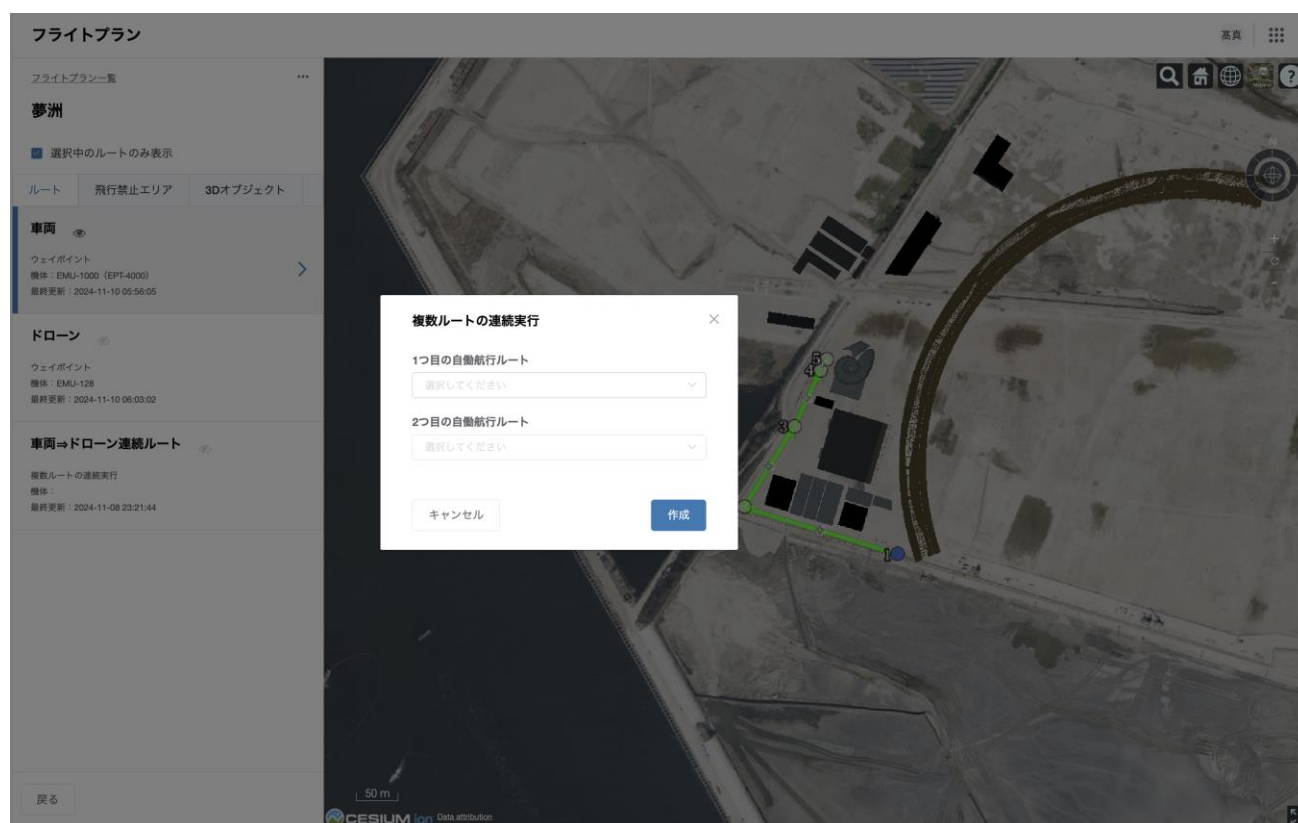


図 4-90 ルート選択イメージ

10. 【SC010】ルートアップロード

● 概要

- ルートアップロード確認画面を選択すると、ポップアップ画面が表示
- ポップアップ画面には、車両・ドローン合計の運航予定時間や運航予定距離などを表示
- 「ルートをアップロード」ボタンを選択すると、車両とドローンにルートがアップロードされる

● 画面イメージ

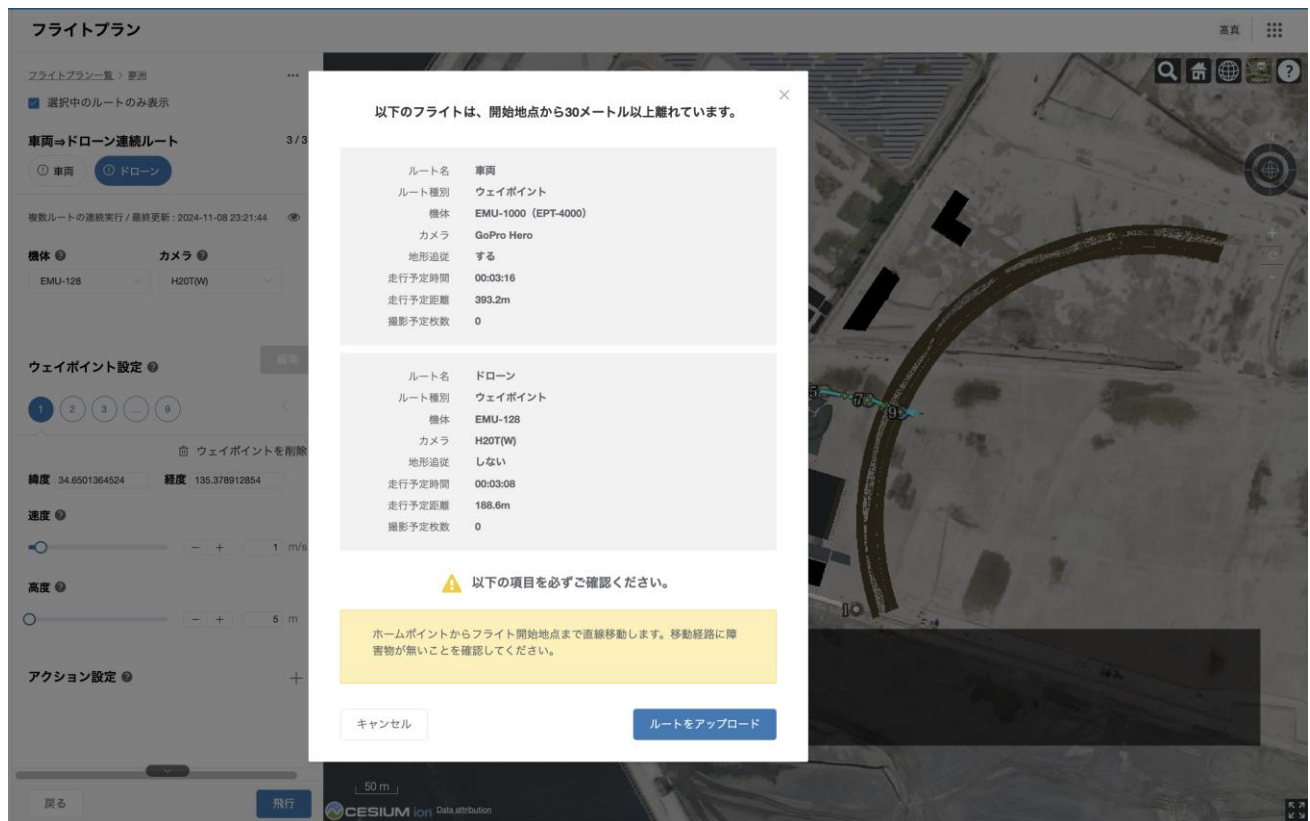


図 4-91 ルートアップロード画面イメージ

11. 【SC011】走行開始画面

● 概要

- 「4-6-3.各画面仕様詳細」の記載と重複するため割愛

● 画面イメージ

- 「4-6-3.各画面仕様詳細」の記載と重複するため割愛

12. 【SC012】モニタリング画面

● 概要

- 「4-6-3.各画面仕様詳細」の記載と重複するため割愛
- 画面イメージ
 - 「4-6-3.各画面仕様詳細」の記載と重複するため割愛

13. 【SC013】車両テレメトリ画面

- 概要
 - 「4-6-3.各画面仕様詳細」の記載と重複するため割愛
- 画面イメージ
 - 「4-6-3.各画面仕様詳細」の記載と重複するため割愛

14. 【SC014】走行完了・飛行開始画面

- 概要
 - 車両走行完了後、車両からの走行完了信号を受け取ると自動的に表示
 - 「2つ目のルートを開始」を押下するとドローンの自律飛行を開始
 - 車両走行後にドローン飛行を行わないと判断した場合は、「キャンセル」を押下することでドローン飛行を行わずに【SC008】ルート統合画面に戻ることが可能
- 画面イメージ

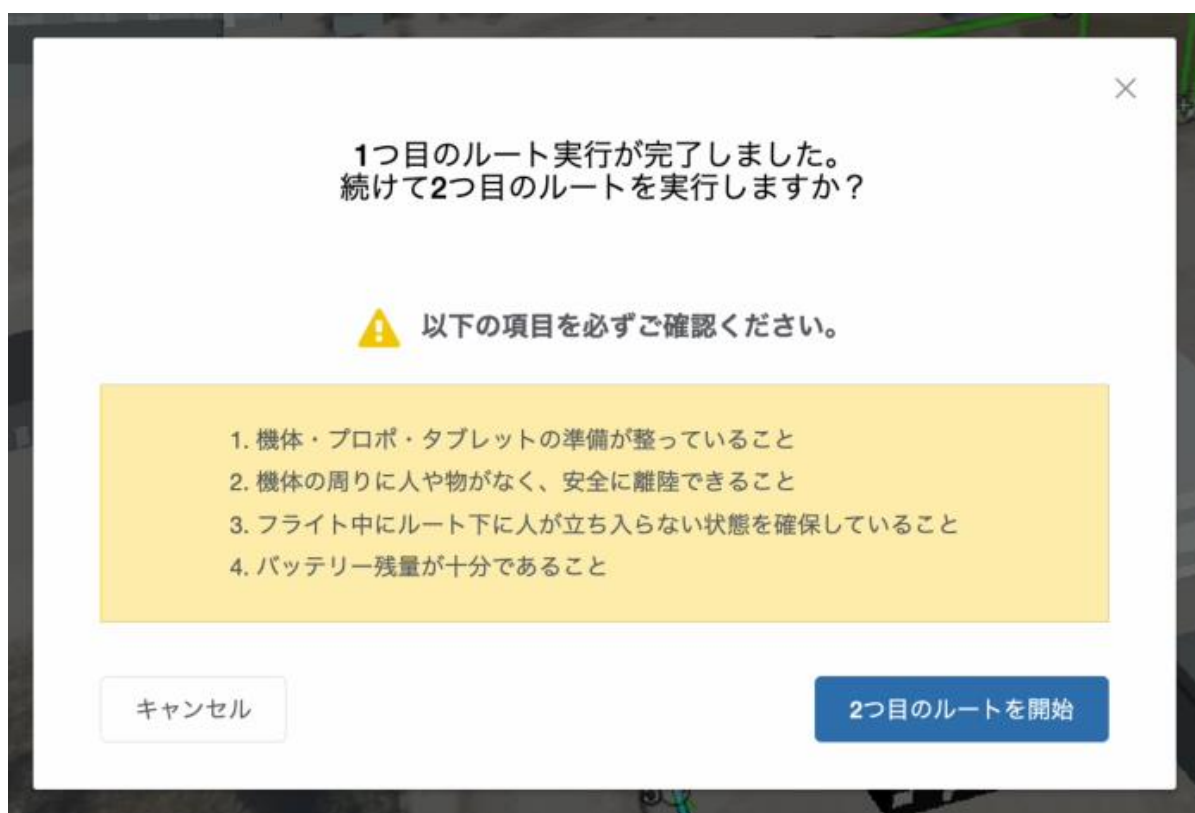


図 4-92 走行完了・飛行開始画面イメージ

15. 【SC015】ドローンテレメトリ画面

- 概要
 - 「4-6-3.各画面仕様詳細」の記載と重複するため割愛
- 画面イメージ
 - 「4-6-3.各画面仕様詳細」の記載と重複するため割愛

16. 【SC016】緊急停止

- 概要
 - 【SC015】ドローンテレメトリ画面右上の飛行機アイコンをクリックすることで、ドローンに指示が可能
 - 「帰還」をクリックで飛行を中断し、ホームポイントに帰還
 - 「一時停止」をクリックで自律飛行を中断し、緊急停止
 - 緊急停止後、「再開」をクリックで自律飛行を再開
- 画面イメージ



図 4-93 緊急停止の画面イメージ

2) 管理者画面

1. 【SC101】動作選択

- 概要

- メイン画面の3点リーダーから、リスト選択で新規3Dオブジェクトの読み込みを選択

- 画面イメージ

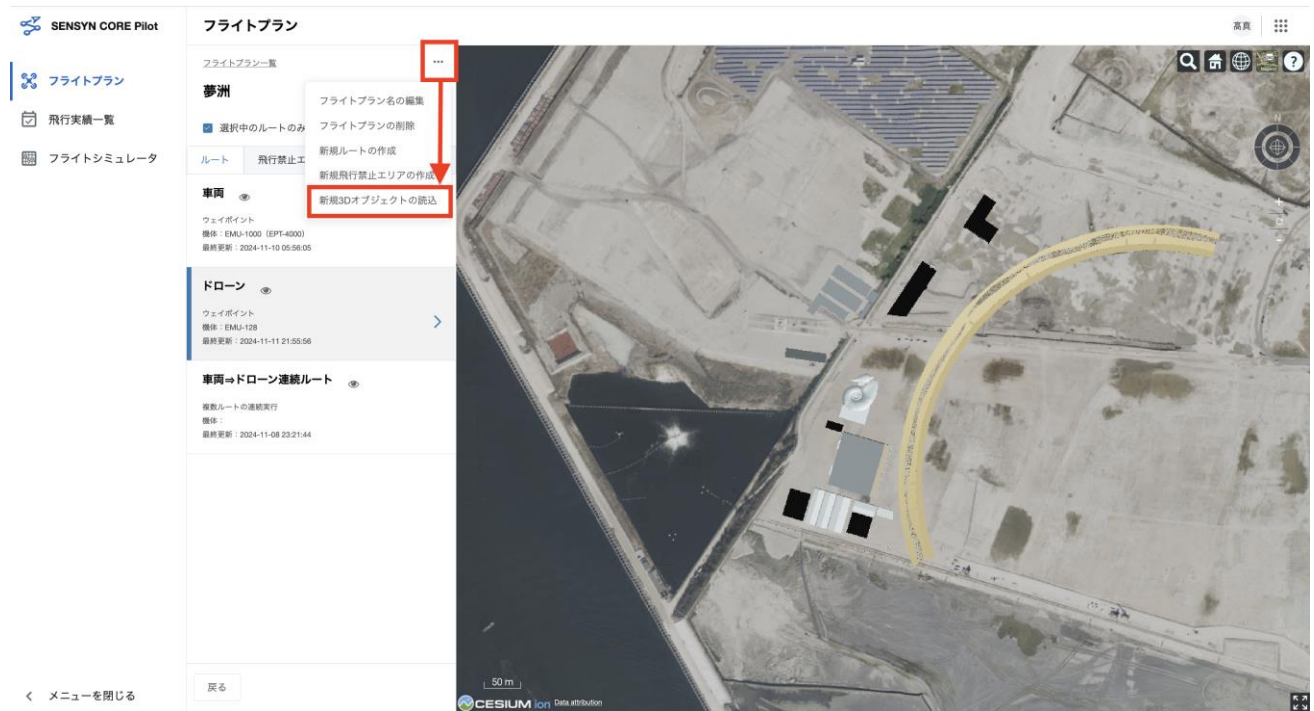


図 4-94 動作選択の画面イメージ

2. 【SC102】新規 3D オブジェクトの登録

- 概要
 - CityGML や BIM から変換したオブジェクトを SENSYN CORE Pilot へインポートする
 - ID 入力で自動的に反映
- 画面イメージ

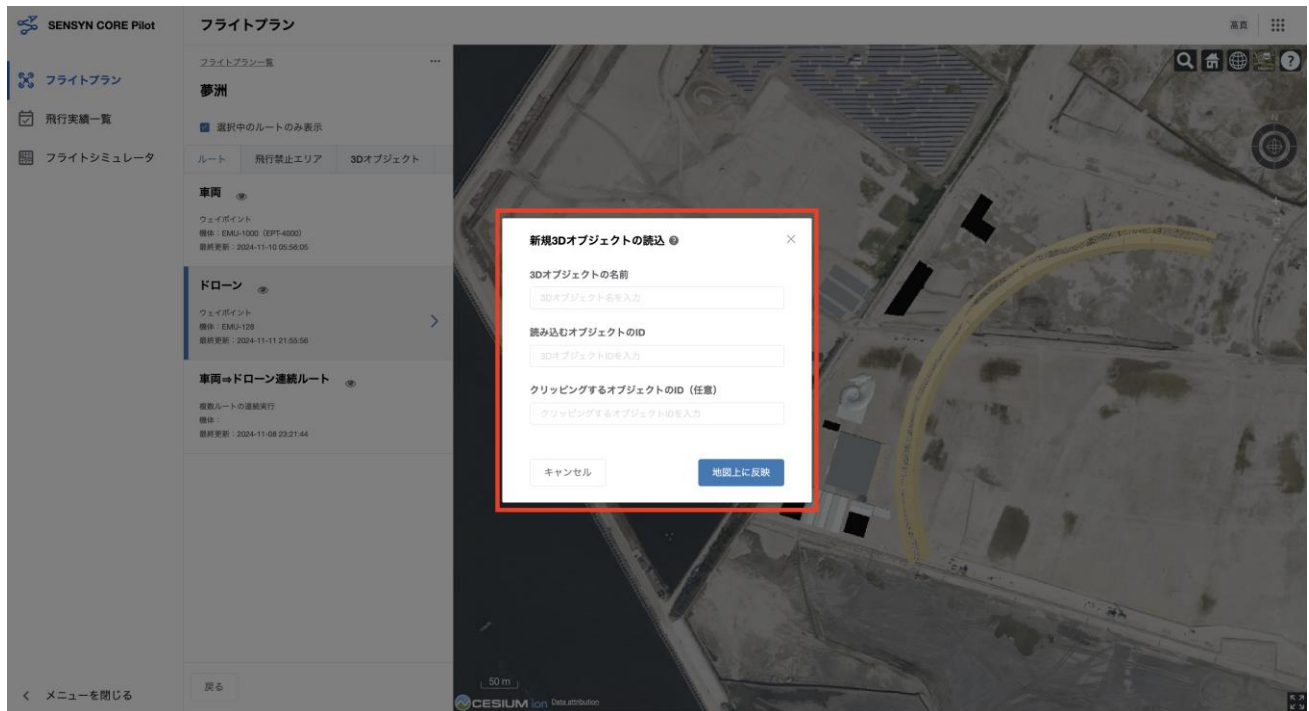


図 4-95 新規 3D オブジェクトの登録イメージ

3. 【SC103】 3D オブジェクト表示

- 概要
 - 地図画面にインポートしたオブジェクトを表示
- 画面イメージ

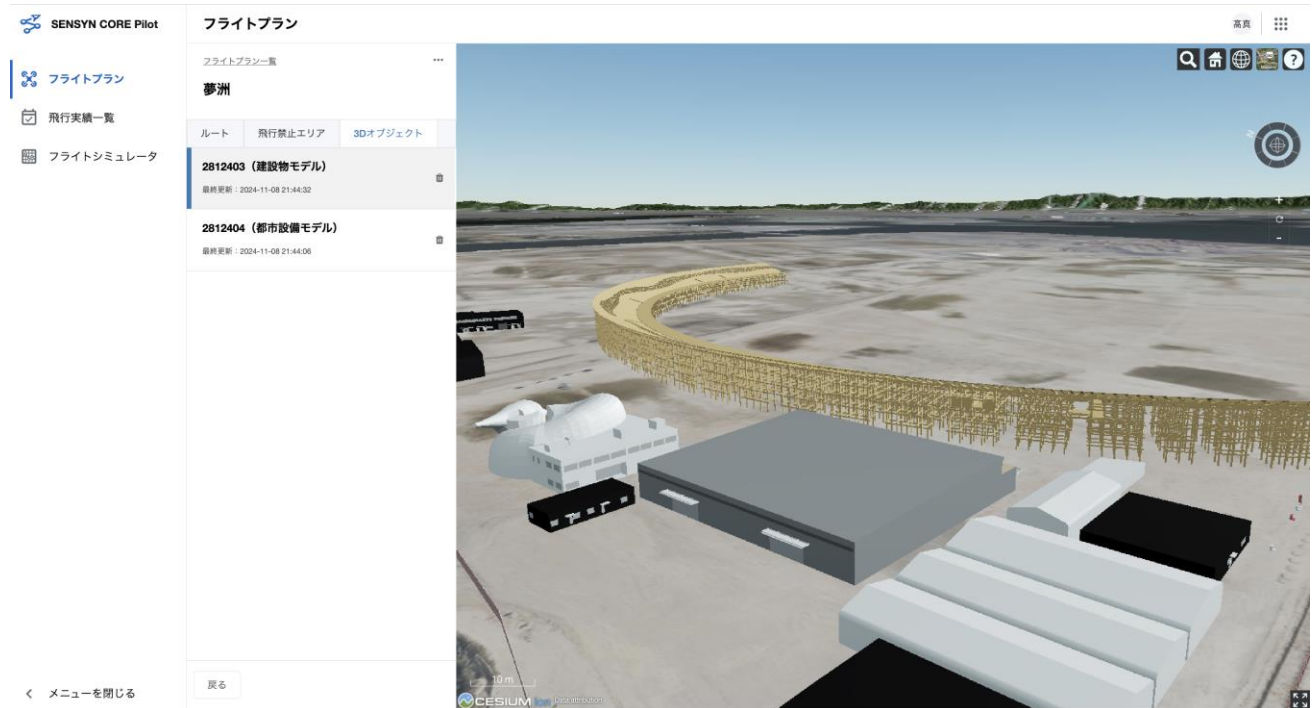


図 4-96 3D オブジェクト表示イメージ

5. システムの非機能要件

5-1. 社会実装に向けた非機能要件

表 5-1 非機能要件一覧

カテゴリ	ID	項目	詳細
可用性	NR001	システムの連続稼働時間	メンテナンス時間等を除き、原則として 24 時間 365 日連続稼働すること
	NR002	安定動作時間	実証環境のため、1 時間以上安定動作すること
	NR003	天候対応	- 車両、ドローンともに晴天、曇天に対応できるよう対策を行うこと - ドローンについては、風速 5m 毎秒まで対応できるよう対策を行うこと
性能・拡張性	NR004	ドローンに搭載する Edge PC	- ドローンに搭載するため、小型かつ軽量であること - Edge PC 内で点群処理などを行う必要があり GPU が必要となることから、グラフィック処理が行えるプロセッサが搭載されていること
	NR005	画面操作性能	一般的に操作しやすい環境にするために、UI で視覚的に分かりやすい状態で入力設定が可能であること
運用・保守性	NR006	セキュリティ	- 対象となるデータ及びシステムの保守のために、通信に暗号化を行うこと - 必要最低限のサービスのみアクセス可能となるよう、ルーター、ファイアウォール等でパケットフィルタリングすること
	NR007	認証	利用ユーザーを特定するためにユーザー名、メールアドレスを設定したアカウントを発行する

1) 【NR001】システムの連続稼働時間

- 本非機能要件を適用するシステム
 - SENSYN CORE Pilot
- 目標値
 - メンテナンス時間等を除き、原則 24 時間 365 日
- 設定理由
 - SENSYN CORE Pilot は既にユーザーに提供されているシステムである。今回の実証実験ではこのシステムに追加改修を行うが、既にサービス利用中のユーザーを対象とした連続稼働が見込まれるため、メンテナンス時間等を除き、原則 24 時間 365 日の連続稼働時間を確保する。

- 評価方法
 - Microsoft Azure 上で下記 (1) ～ (4) の監視を行う
 - (1) 死活監視
 - (2) プロセス監視
 - (3) ネットワーク監視
 - (4) リソース使用状況監視

- 2) 【NR002】安定動作時間
 - 本非機能要件を適用するシステム
 - SENSYN CORE Pilot
 - 目標値
 - 実証環境のため、1 時間以上安定動作すること
 - 設定理由
 - ラストワンマイル輸送に最低限必要な時間を想定し、設定
 - 評価方法
 - システム稼働テストを複数回行い、システムが安定して稼働することを確認する

- 3) 【NR003】天候対応
 - 本非機能要件を適用するシステム
 - 自動搬送車両用システム
 - ELAS
 - 目標値
 - 車両、ドローンともに晴天、曇天に対応
 - ドローンについては、上空の風速で 5m 毎秒まで対応
 - 設定理由
 - 従来の業務フロー上、ドローンは雨天又は強風などの荒天では飛行しない運用がなされている。そのため、今回の実証実験でも晴天又は曇天で、かつ上空の風速については安全にドローン飛行が行える風速 5m 毎秒以下での運航可用性を確保する
 - 評価方法
 - 条件を満たす天候下でシステム稼働テストを複数回行い、システムが安定して稼働することを確認する

- 4) 【NR004】ドローンに搭載する Edge PC
 - 本非機能要件を適用するシステム
 - ELAS
 - 目標値
 - ドローンに搭載可能な小ささ、重さ
 - グラフィック処理が行えるプロセッサが搭載されている

- 設定理由
 - ドローンに搭載するため、小型かつ軽量であることが必要であるため
 - Edge PC 内で点群処理などを行う GPU が必要となるため、グラフィック処理が行えるプロセッサが搭載されている必要がある
 - 評価方法
 - ドローン飛行テストを複数回行い、安定して飛行することと、点群処理が行えることを確認する
- 5) 【NR005】画面操作性
- 本非機能要件を適用するシステム
 - SENSYN CORE Pilot
 - 目標値
 - UI で視覚的に分かりやすい状態で入力設定が可能
 - 設定理由
 - 従来の業務フロー上、専門的な知識を持たないユーザーがシステムを操作することがある。今回の実証実験においても、社会実装を目指し、一般的なユーザーが操作しやすいシステムとするため UI で視覚的に分かりやすい状態で入力設定が可能であることが求められる
 - 評価方法
 - システム稼働テストを複数回・複数名で行い、視覚的に分かりやすい入力設定が行えることを確認する
- 6) 【NR006】セキュリティ
- 本非機能要件を適用するシステム
 - SENSYN CORE Pilot
 - 自動搬送車両用システム
 - ELAS
 - 目標値
 - 通信に暗号化を行う
 - 必要最低限のサービスのみアクセス可能となるよう、ルーター、ファイアウォール等でパケットフィルタリングする
 - 設定理由
 - 対象となるデータ及びシステムの保護。
 - 評価方法
 - 許可されていないアカウントからのアクセスができないことを確認する
- 7) 【NR007】認証
- 本非機能要件を適用するシステム
 - SENSYN CORE Pilot
 - 目標値

- 利用ユーザーを特定するためにユーザー名、メールアドレスを設定したアカウントを発行する
- 設定理由
 - 対象となるデータ及びシステムの保護
- 評価方法
 - 許可されていないアカウントからのアクセスができないことを確認する

5-2. 実証観点での非機能要件

表 5-2 非機能要件一覧

カテゴリ	ID	項目	詳細
可用性	NR001	システムの連続稼働時間	● 水平・垂直方向の自動運航を複数回実施するため、原則 5 時間連続稼働すること
	NR002	安定動作時間	● 実証環境のため、1 時間以上安定動作すること
	NR003	天候対応	● 車両、ドローンともに晴天、曇天に対応できるよう対策を行うこと ● ドローンについては、風速 5m 毎秒まで対応できるよう対策を行うこと
性能・拡張性	NR004	ドローンに搭載する Edge PC	● ドローンに搭載するため、小型かつ軽量であること ● Edge PC 内で点群処理などを行う必要があり GPU が必要となることから、グラフィック処理が行えるプロセッサが搭載されていること
	NR005	画面操作性能	● 一般的に操作しやすい環境にするために、UI で視覚的に分かりやすい状態で入力設定が可能であること

1) 【NR001】システムの連続稼働時間

- 本非機能要件を適用するシステム
 - SENSYN CORE Pilot
- 目標値
 - メンテナンス時間等を除き、原則 5 時間
- 設定理由
 - SENSYN CORE Pilot は既にユーザーに提供されているシステムであり、また今回の実証でも複数回の自動運航を実施するため、原則 5 時間の連続稼働時間を確保する
- 評価方法
 - Microsoft Azure 上で下記 (1) ～ (4) の監視を行う
 - (1) 死活監視
 - (2) プロセス監視

- (3) ネットワーク監視
- (4) リソース使用状況監視

2) 【NR002】安定動作時間

- 本非機能要件を適用するシステム
 - SENSYN CORE Pilot
- 目標値
 - 実証用の環境を切り出すため運用は不要であり、定義はなし
- 設定理由
 - SENSYN CORE Pilot は既にユーザーに提供されているシステムであるため、既存ユーザーに与える影響を最小限にするため、本実証実験専用の環境を切り出す。そのため、運用は不要
- 評価方法
 - 不要

3) 【NR003】天候対応

- 本非機能要件を適用するシステム
 - 自動搬送車両用システム
 - ELAS
- 目標値
 - 車両、ドローンともに晴天、曇天に対応
 - ドローンについては、上空の風速で 5m 毎秒まで対応
- 設定理由
 - 従来の業務フロー上、ドローンは雨天又は強風などの荒天では飛行しない運用がなされている。そのため、今回の実証実験でも晴天又は曇天で、かつ上空の風速については安全にドローン飛行が行える風速 5m 毎秒以下での運航可用性を確保する
- 評価方法
 - 条件を満たす天候下でシステム稼働テストを複数回行い、システムが安定して稼働することを確認する

4) 【NR004】ドローンに搭載する Edge PC

- 本非機能要件を適用するシステム
 - ELAS
- 目標値
 - ドローンに搭載可能な小ささ、重さ
 - グラフィック処理が行えるプロセッサが搭載されている
- 設定理由
 - ドローンに搭載するため、小型かつ軽量であることが必要であるため
 - Edge PC 内で点群処理などを行う GPU が必要となるため、グラフィック処理が行えるプロセッサ

ーが搭載されている必要がある

- 評価方法

- ドローン飛行テストを複数回行い、安定して飛行することと、点群処理が行えることを確認する

5) 【NR005】画面操作性能

- 本非機能要件を適用するシステム

- SENSYN CORE Pilot

- 目標値

- UI で視覚的に分かりやすい状態で入力設定が可能

- 設定理由

- 従来の業務フロー上、専門的な知識を持たないユーザーがシステムを操作することがある。今回の実証実験においても、社会実装を目指し、一般的なユーザーが操作しやすいシステムとするため UI で視覚的に分かりやすい状態で入力設定が可能であることが求められる。

- 評価方法

- システム稼働テストを複数回・複数名で行い、視覚的に分かりやすい入力設定が行えることを確認する

6. 品質

6-1. 機能要件の品質担保

表 6-1 機能要件の品質担保方針

対象プロセス/ サブシステム	品質評価項目	目標値	期間の単位	アクティビティ
3D 都市モデルと BIM のマージ	自動搬送車両走行及びドローン飛行に資する精度	● CityGML 基準 /BIM 基準	2024/5～10 月	● ソフトウェアによるシミュレーション
GCS アプリによる 自動搬送車両連携機能構築	メッセージの送受信が意図通りか	● 複数回実施し、エラー発生回数/試行回数を計測	2024/7～9 月	● 単体テストによる検証
自動搬送車両・シミュレーション走行再現性検証	指定ルートの走行をシミュレーションで再現	● 複数回実施し、ゴール到達回数/試行回数を計測	2024/7～11 月	● ソフトウェアによるシミュレーション
ドローン・シミュレーション飛行再現性検証	指定ルートの飛行をシミュレーションで再現	● 複数回実施し、ゴール到達回数/試行回数を計測	2024/7～11 月	● ソフトウェアによるシミュレーション
飛行再現性検証 (夢洲エリアでの実証実験)	指定ルートの飛行を実環境で再現	● 達成/未達成	2024/10～12 月	● 実環境での飛行

6-2. 非機能要件の品質担保

表 6-2 非機能要件の品質担保方針

対象項目	品質評価項目	目標値	期間の単位	アクティビティ
SENSYN CORE Pilot	システムの連続稼働時間	● メンテナンス時間等を除き、原則 24 時間 365 日	2024/11～12 月	● Microsoft Azure 上で、5 章で定めた通りの監視を実施
	安定動作時間	● 実証環境のため、1 時間以上安定動作すること	2024/11～12 月	● なし
	セキュリティ	● 通信に暗号化を行う ● 必要最低限のサービスのみアクセス可能となるよう、ルーター、ファイアウォール等でパケットフィルタリングする	2024/11～12 月	● 許可されていないアカウントからのアクセスができないことを確認する証を実施
	認証	● 利用ユーザーを特定するためにユーザー名、メールアドレスを設定したアカウントを発行する	2024/11～12 月	● 許可されていないアカウントからのアクセスができないことを確認する検証を実施
	UI	● ユーザーヒアリング（使いやすさ）	2024/12 月～2025/1 月	● ヒアリング
自動搬送車両用システム、ELAS	天候対応	● 晴天、曇天に対応 ● ドローンについては、上空の風速で 5m 毎秒まで対応	2024/7～12 月	● 条件を満たす天候下でシステム稼働検証を複数回行う

	自己位置推定性能	● 3D 都市モデル由来の点群マップにより、自己位置推定が行える	2024/7～12 月	● システム稼働テストを複数回行い、自己位置推定が行えるか検証
	点群データ作成	● 3D 都市モデルから点群マップを作成できること由来の点群マップにより、自己位置推定が行える	2024/7～12 月	● システム稼働テストを複数回行い、自己位置推定が行えるか検証
ELAS	ドローンに搭載する Edge PC	● ドローンに搭載可能な小ささ、重さ ● グラフィック処理が行えるプロセッサが搭載されている	2024/7～12 月	● ドローン飛行テストを複数回行い、安定して飛行することと、点群処理が行えるか検証

7. 実証技術の機能要件の検証

7-1. RTK 測位データを活用した車両の自己位置精度向上の検証

7-1-1. 検証目的

- RTK 測位による自己位置補完が有効であることを、マッチングスコア改善度を算出することによって検証する

7-1-2. KPI

表 7-1 KPI 一覧

評価指標・KPI			No.	目標値	目標値の設定理由	検証方法
KPI	比較対象 スコア	モデル整備 状況				
マッチングスコア	-	片側整備/ 両側整備	1	2.0 以上	● 3D 都市モデルの整備状況に関わらず、RTK 測位による補完をすることで、自動搬送車両が自動運行できるマッチングスコア 2.0 以上を設定	● PCLG マップ+ RTK 測位利用時のマッチングスコアより算出
マッチングスコア改善度	リアルマップ	両側整備	2	0%以上	● 3D 都市モデル両側整備エリアでは、PCLG マップによるマッチングスコアが一定レベルを維持し、RTK 測位による改善は小さい見込み ● RTK 測位の補完によって、マッチング精度が下がることが無いように設定	● 各種マッチングスコアより算出
		片側整備	3	30%以上	● 3D 都市モデル片側整備エリアでは、PCLG マップによるマッチングスコアが低く、RTK 測位による改善が大きい見込み ● 同等機材の過去検証結果より、使用機器のハード構成ではリアルマップとのマッチングスコアが最小 8.0 程度となる想定 ● PCLG マップによる自己位置推定がロスト(<2.0)した場合に、RTK 測位の補完によって復帰できる水準に設定	

各指標定義

本実証では RTK 測位の補完によるマッチング精度の改善を評価するための指標としてマッチングスコア改善度を定義した。マッチングスコア改善度は、RTK 測位の補完によって、PLCG マップ利用時のマッチングスコアがどの程度リアルマップ利用時のマッチングスコアとの差を縮めたかを示す指標である。

$$\text{マッチングスコア改善度}[\%] = \frac{A-B}{A} \times 100$$

A = [リアルマップ利用時のマッチングスコア] - [PLCG マップ利用時のマッチングスコア]

B = [リアルマップ利用時のマッチングスコア] - [PLCG マップ+RTK 測位利用時のマッチングスコア]

- マッチングスコア

- LiDAR のスキャンデータと点群地図の適合度（マッチング精度）を示すスコアである
- スコアは最小 0.0~の数値で示され、この値が大きいほど正確にスキャンマッチングされると言える。本実証ではリアルマップのマッチングスコアについて、最大 10.0 程度の値が見込まれるため最小 0.0~最大 10.0 で評価する
- Autoware では、NDT（Normal distributions transform）実装を利用しており、スコアはスキャン後の 1 点あたりの平均適合スコアとして計算される。この値は、スキャンポイントがマップの正規分布（ガウス分布）にどれだけ適合しているかを数値化したものであり、NDT による自己位置推定の信頼性を示す

表 7-2 マッチングスコア閾値

マッチングスコア	自動運行への影響
8.0 以上	10.0km/h 以上の自動運行が可能
2.0 以上、8.0 未満	5.0km/h 以上の自動運行が可能
2.0 未満	自己位置ロストし自動運行不可

7-1-3. 検証方法と検証シナリオ

1) 検証エリア

- 3D 都市モデルが両側に整備されている「両側整備エリア」と片側にしか整備されていない「片側整備エリア」に分類した。それぞれのエリアを「リング付近」「倉庫前（前半）」「倉庫前（後半）」「カーブエリア」「フェンス前」「ベース付近」に分け、以下の図で示す
 - 両側整備エリアと比べて、片側整備エリアでは点群マッチング精度が下がり、RTK-GNSS 測位の優位性が高くなると想定し検証した
 - ルート外周を囲うように都市設備モデル（ガードレールやフェンス等、図 7-1 の黄緑で表示した部分）が整備されており、両側整備エリアはその反対側に建築物モデルが整備されている箇所、片側整備エリアは反対側に建築物モデルが整備されていない箇所とした

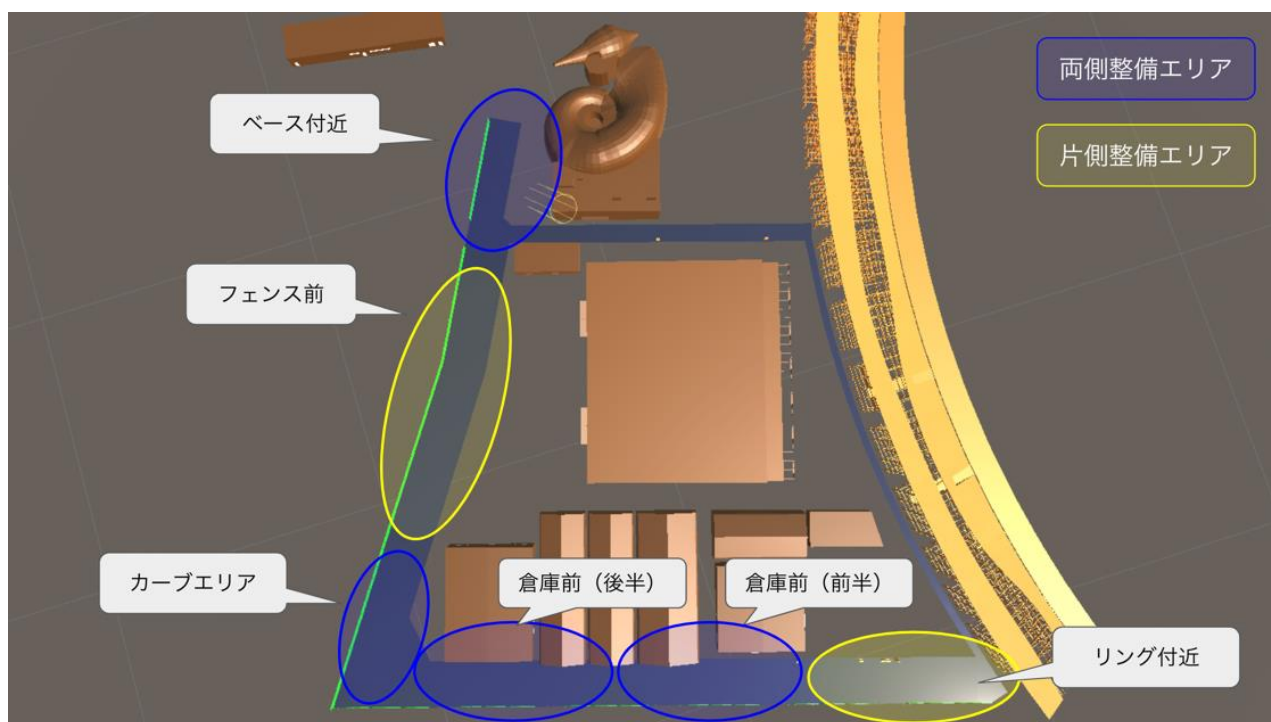


図 7-1 3D 都市モデル整備エリア（真上から見た図）

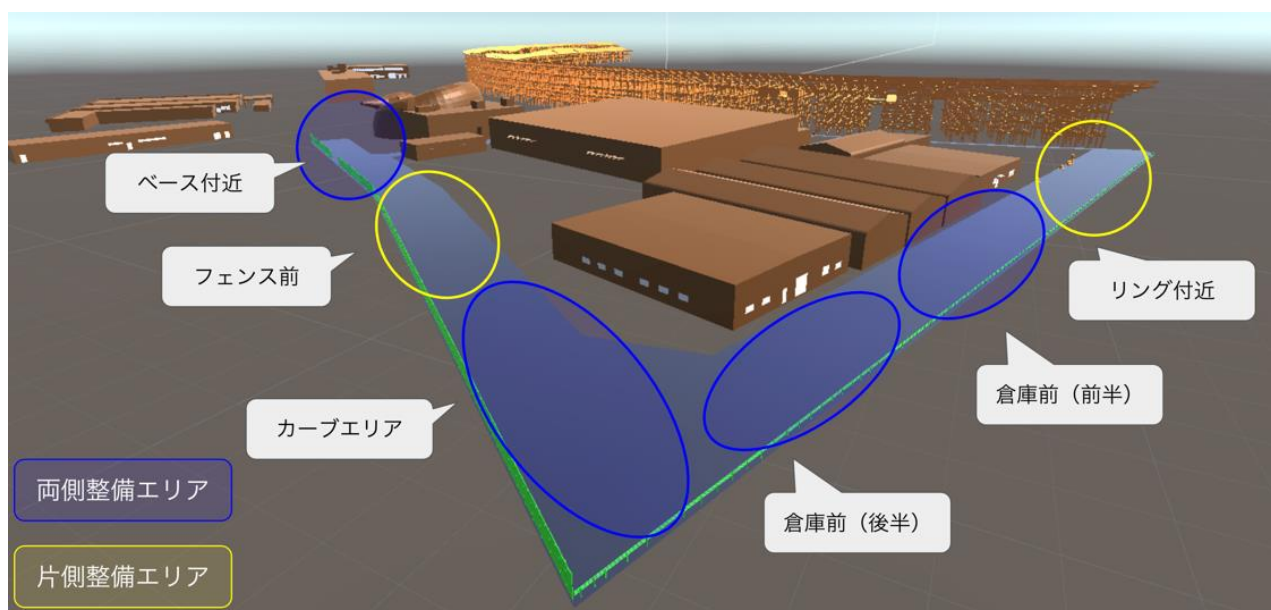


図 7-2 3D 都市モデル整備エリア（斜め上から見た図）

2) 点群マップ比較

● 検証で利用する 2 つの点群マップを、以下の図で示す

- 事前走行により取得した点群マップを「リアルマップ」、PLATEAU PointCloud Generator 経由で生成された点群マップを「PCLG マップ」と呼ぶ

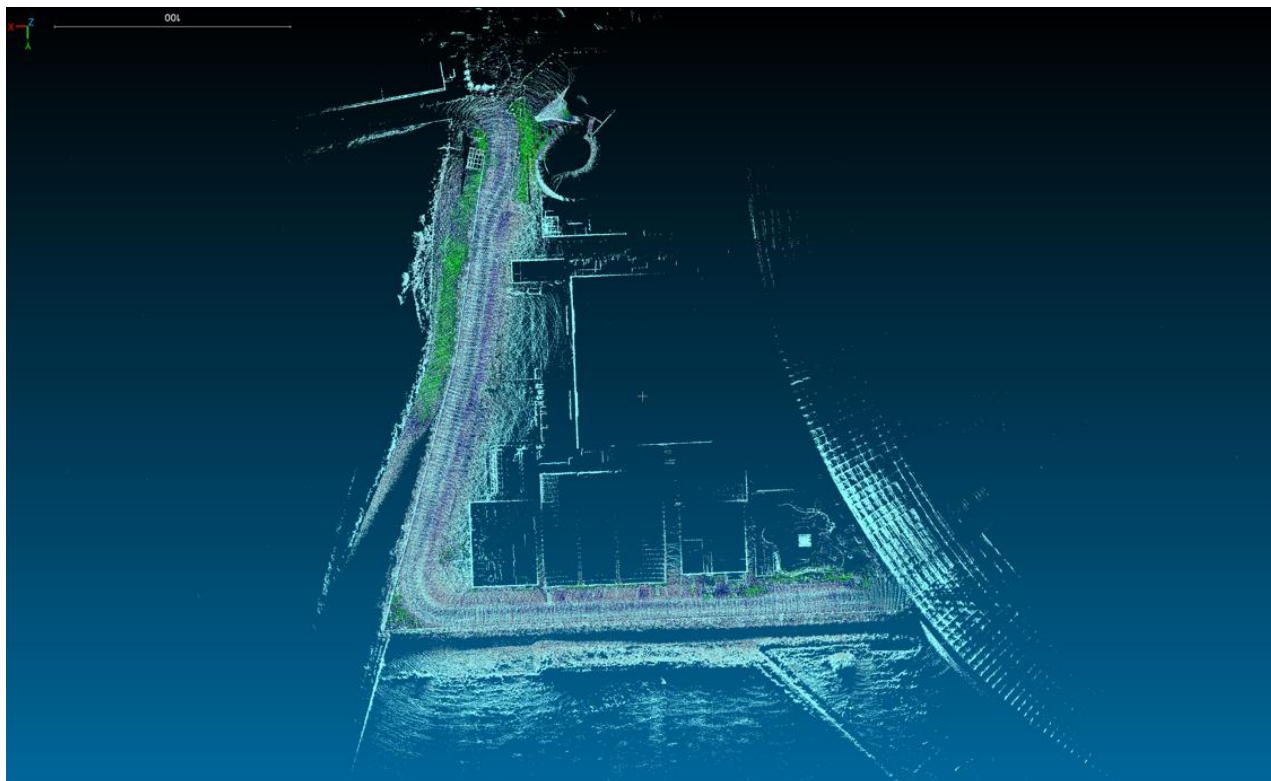


図 7-3 事前走行により取得したリアルマップ

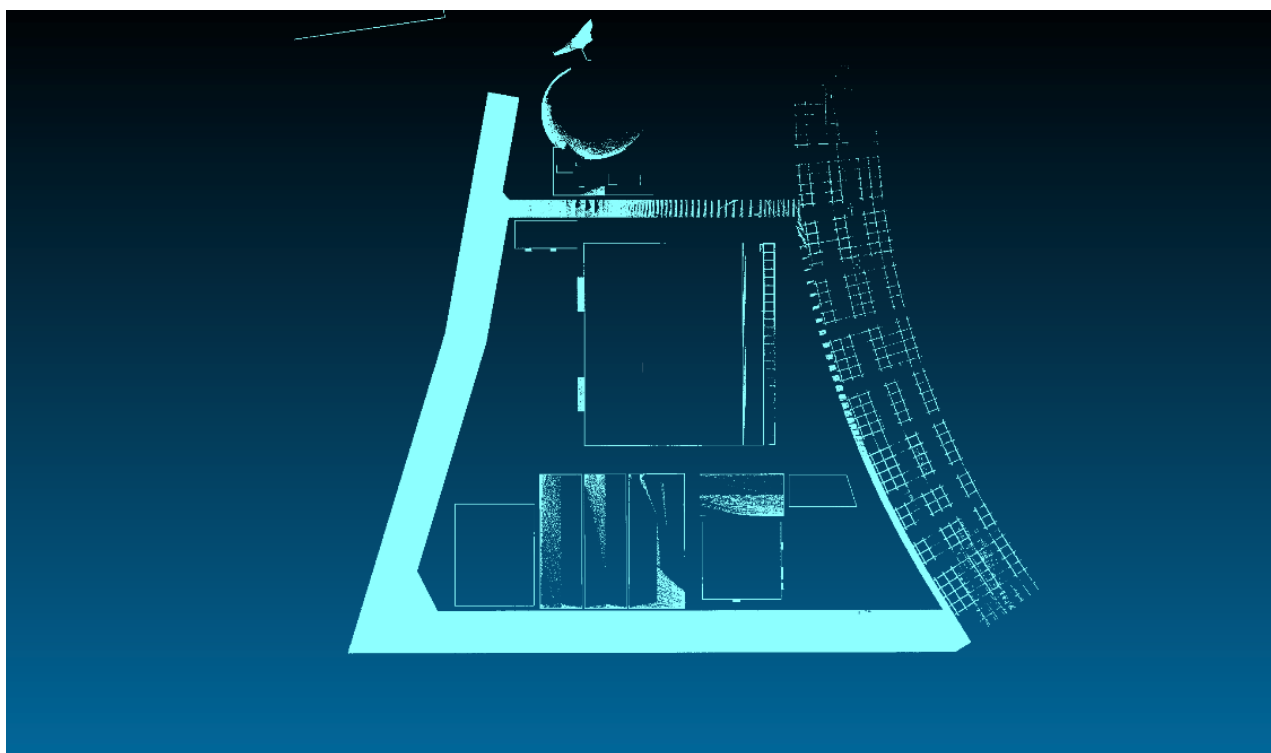


図 7-4 PLATEAU PointCloud Generator により生成された PCLG マップ

3) 3D 都市モデル整備エリアでの精度比較

- 自己位置推定の精度割合を、下記の 2 パターンで検証する
 - リアルマップと PCLG マップを用いた場合の自己位置測位の精度を検証
 - PCLG マップを利用した場合の RTK 測位による優位性を検証

表 7-3 検証シナリオ一覧（利用マップ別）

検証エリア	ベースマップ	比較対象マップ
両側整備エリア	リアルマップ	PCLG マップ+RTK 測位
	PCLG マップ	PCLG マップ+RTK 測位
片側整備エリア	リアルマップ	PCLG マップ+RTK 測位
	PCLG マップ	PCLG マップ+RTK 測位

4) 実証実験の様子

- 実証実験の様子を下記の図に示す
 - 遠隔監視ベースでは SENSYN CORE Pilot でのテレメトリー確認、歩行者や通行車両の監視を行った
 - 緊急時には車両を手動操作で切り替えられるように、テストドライバーが乗車した



図 7-5 実証実験の様子（遠隔監視ベース付近）



図 7-6 実証実験の様子（カーブエリアの車両走行風景）



図 7-7 実証実験の様子（フェンス前の車両走行風景）

7-1-4. 検証結果

検証の結果、全てのエリアにおいて、自動運行に必要なマッチングスコア 2.0 以上を達成することができた。また、6 つの内 2 つのエリアでは RTK-GNSS 測位によって補完された自己位置推定精度が点群マッチングの精度を大きく上回り、特定条件において RTK-GNSS 測位による補完が有効であることが分かった。マッチングスコア改善度はそれぞれ、3D 都市モデル整備時点からの工事進捗などにより PCLG マップと現実の建築物に大きな差異が発生した箇所を含むエリアでは 30.5%、同じ形状の施設が並び自己位置をロストしやすい箇所を含むエリアでは 11.9%であり、優位性が顕著に現れた。一方で、当初 RTK-GNSS 測位の優位性が高いと想定した「片側整備エリア」では優位性を示す結果が得られなかった。これは、片側整備エリアであってもフェンスなどの都市設備モデルの特徴点が多く取得できたことで、点群マッチングによる自己位置推定が十分に機能したことに起因する。

このことから、RTK-GNSS 測位による補完は、点群マッチングにおいて建物単位の大きいノイズが発生する場合や、同じ形状が続き固有の特徴点を取得することが難しい場合において、大きな改善を見込めることが示された。

- マッチングスコア（PCLG マップ+RTK 測位利用時）

全てのエリアにおいて自動運行に必要なマッチングスコア 2.0 以上を達成した。特に、リング付近エリアでは PCLG マップのみでは 2.0 を下回るマッチングスコアとなっていたところ、RTK 測位による補完をすることで、目標である 2.0 以上を維持することができた。

表 7-4 マッチングスコアの目標値比較表

赤セル：KPI 達成

青セル：KPI 未達

走行エリア	モデル整備状況	目標値	結果	考察
リング付近	片側整備	2.0 以上	2.80	リング付近では PCLG マップの精度が大きく低下した。建設中のパビリオンが 3D 都市モデルで再現されていなく、現地との差異が大きかったことが起因する。PCLG マップ単体では自己位置をロストし（マッチングスコア 0.41）、自動運転は出来なかったが、RTK 測位で自己位置情報を補完することで、自動運転で走行可能なマッチングスコア 2.0 以上を達成することができた。
倉庫前（前半）	両側整備	2.0 以上	3.28	倉庫前の前半エリアでは、同じ形状の建物が連続したことによって、点群マッチングによる自己位置推定精度が低下し、RTK 測位無しの状態では、点群マッチングのスコアが上昇するまでに時間がかかった。RTK 測位ありの場合は、マッチングは安定して自動運転ができた。
倉庫前（後半）	両側整備	2.0 以上	3.48	倉庫前の後半エリアでは、前半エリアよりも路上駐車の影響が少なく、点群マッチングによる自己位置推定精度が向上した。

半)				車や、建築資材が設置されておらず、点群マッチングのスコアが上昇し、安定したマッチングスコアを維持した。
カーブエリア	両側整備	2.0 以上	4.20	カーブエリアでは、フェンス、倉庫の角の特徴点を取ることができ、安定したマッチングスコアを維持した。
フェンス前	片側整備	2.0 以上	3.85	フェンス前では地物が少なく、PCLG マップ利用時のマッチングスコアが低下すると想定したが、フェンスの特徴点を効果的に得ることができた。これにより点群マッチングの精度が高い値で推移し、安定したマッチングスコアを維持した。
ベース付近	両側整備	2.0 以上	2.78	ベース付近では、パビリオン、フェンスの 3D 都市モデルが両側に整備されており、点群マッチングの精度が高い値で推移し、安定したマッチングスコアを維持した。他のエリアと比較してマッチングスコアが低い値となった要因としては、3D 都市モデルに再現されていなかった倉庫や停留所などの地物が設置されたこと、パビリオンが球状で特徴点を捉えにくかったことが起因する。

表 7-5 マッチングスコア比較表

エリア	PCLG マップのみ	PCLG マップ+RTK 補正あり
リング付近	0.41	2.80
倉庫前（前半）	2.55	3.28
倉庫前（後半）	3.39	3.48
カーブ	4.16	4.20
フェンス前	3.87	3.85
ベース付近	2.81	2.78

● マッチングスコア改善度（PLCG マップ+RTK 測位利用時）

当初想定していた片側整備エリアにおける RTK-GNSS 測位の優位性は認められなかった。フェンス等の一定以上の高さがある都市設備の特徴点を捉えることによって、点群マッチングによる自己位置推定精度が高い値で推移したことが主な要因として考えられる。一方で、倉庫前（前半）エリアのように、同じ形状の建物が連続する場合に点群マッチング精度が低下し、RTK-GNSS 測位による補完が有効だということが分かった。

表 7-6 マッチングスコア改善度の目標値比較表

赤セル：KPI 達成	青セル：KPI 未達
------------	------------

走行エリア	モデル整備状況	目標値	結果	考察
リング付近	片側整備	30 %以上	30.5%	リング付近では他の走行エリアと比べ、リアルマップ、PCLG マップ双方でマッチングスコアが低下した。点群マッチングのスコアが低くなった要因としては、建設中のパビリオンがあり、現実との差異が大きかったことが起因する。RTK 測位補完によりマッチングスコアを大きく改善し、目標値である 30%の数値を上回る結果となった。
倉庫前（前半）	両側整備	0 %以上	11.9%	倉庫前（前半）エリアでは、同じ形状の建物が連続したことによって、点群マッチングによる自己位置推定精度が低下した。RTK 測位補完によってマッチングスコアが改善し、想定よりも大きな精度向上が見られた。
倉庫前（後半）	両側整備	0 %以上	1.6%	倉庫前（後半）エリアでは、点群マッチングが高精度で推移し、想定した通り大きな改善は認められなかった。
カーブエリア	両側整備	0 %以上	0.8%	カーブエリアでは、点群マッチングが高精度で推移し、想定した通り大きな改善は認められなかった。
フェンス前	片側整備	30 %以上	-0.4%	フェンス前ではフェンスしかモデルがなく、マッチングスコアは下がるかと思われたが、フェンスの 3D 都市モデルの特徴点が正確に取れ、点群マッチングが高精度で推移した。これにより、想定に反して RTK 測位による改善は認められず、目標である 30%以上には至らなかった。
ベース付近	両側整備	0 %以上	-0.5%	ベース付近では、点群マッチングが高精度で推移し、想定した通り大きな改善は認められなかった。RTK 測位補完によってマッチングスコアが若干下落し目標の 0%以上を達成できなかったが、誤差範囲であり自動運行への影響はない。RTK 測位を利用することでマッチングスコアが必ず上昇するわけではなく、自己位置推定の計算量の増加により、マッチングスコアが微減するという傾向が得られた。

(再掲)

$$\text{マッチングスコア改善度}[\%] = \frac{A-B}{A} \times 100$$

A = [リアルマップ利用時のマッチングスコア] - [PCLG マップ利用時のマッチングスコア]

B = [リアルマップ利用時のマッチングスコア] - [PCLG マップ+RTK 測位利用時のマッチングスコア]

表 7-7 マッチングスコア、マッチングスコア改善度比較表

エリア	PCLG マップのみ	PCLG マップ+RTK 補正あり	比較対象（リアルマ ップ）	マッチングスコア改 善度
リング付近	0.41	2.80	8.25	30.5%
倉庫前（前半）	2.55	3.28	8.67	11.9%
倉庫前（後半）	3.39	3.48	9.17	1.6%
カーブ	4.16	4.20	9	0.8%
フェンス前	3.87	3.85	8.44	-0.4%
ベース付近	2.81	2.78	8.53	-0.5%

- 各指標詳細

マッチングスコア改善度の算出にあたり、利用した指標を以下に示す。

各指標定義（再掲）

$$\text{マッチングスコア改善度}[\%] = \frac{A-B}{A} \times 100$$

A = [リアルマップ利用時のマッチングスコア] - [PCLG マップ利用時のマッチングスコア]

B = [リアルマップ利用時のマッチングスコア] - [PCLG マップ+RTK 測位利用時のマッチングスコア]

[リアルマップを用いた自己位置精度]

事前走行により取得したマップとの点群マッチングを行うため、マッチング精度は常に 8.0~10.0 程度の高い値で推移した。

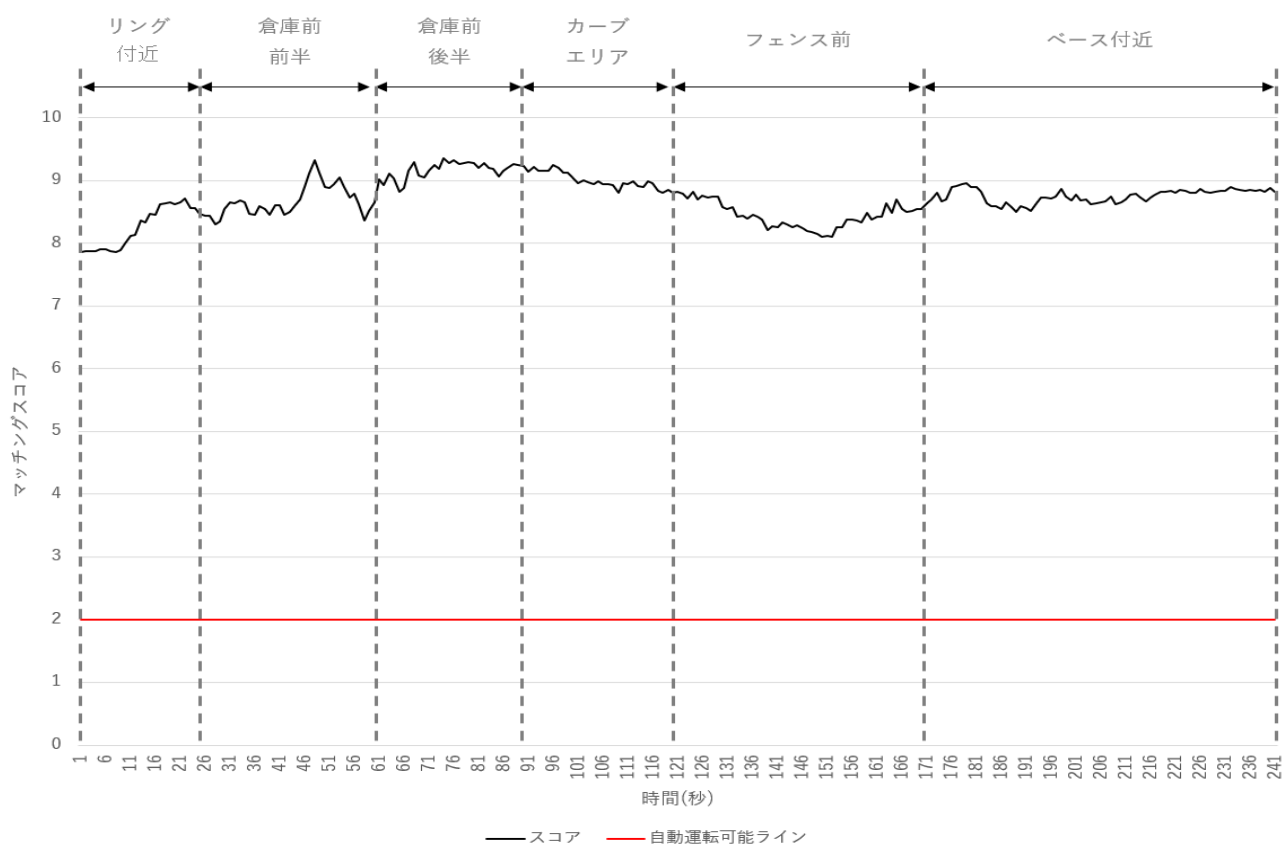


図 7-8 マッチングスコア推移グラフ（リアルマップ利用時）



図 7-9 マッチングスコアヒートマップ（リアルマップ利用時）

リング付近でマッチングスコアが低下した原因

- 点群マップを取得した時にはリング付近ではパビリオンが建築中であった
- 点群マップを取得した時には建物がなかったが、検証期間に建設が進んだため、マッチングスコアが低下した

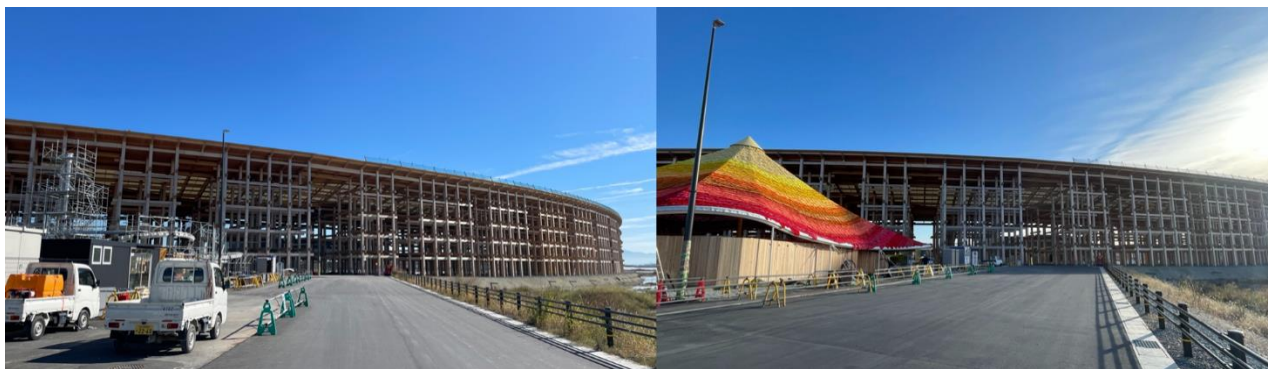


図 7-10 リング付近の様子（左：11/19 撮影 右：12/7 撮影）

フェンス前でマッチングスコアが減少した原因

- フェンス前の通りには、フェンスの側に荷物の資材が山積みとなっていた
- 点群マップを取得した時にはこの資材がなかったため、マッチングスコアが減少した



図 7-11 フェンス前の様子（左：11/19 撮影 右：12/7 撮影）

[PCLG マップを用いた自己位置精度] (RTK 測位無し)

スタート地点であるリング付近でマッチングスコアが 2.0 を下回り、点群マッチングによる自己位置推定がロストした。倉庫前（前半）からマッチングスコアが上昇し、自動運転が可能な閾値を超え、カーブエリアで最も高いスコアとなった。

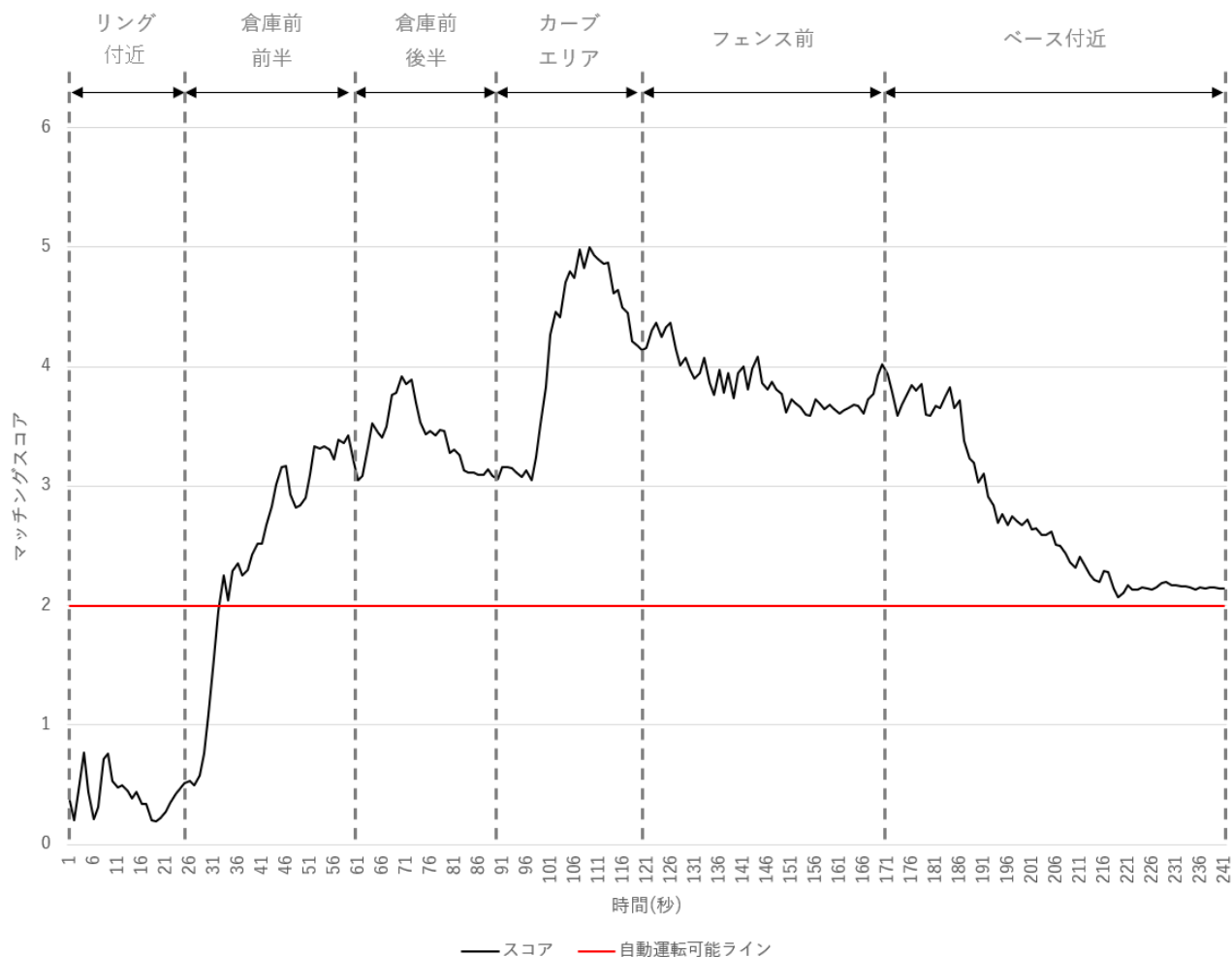


図 7-12 マッチングスコア推移グラフ (PCLG マップ利用時)



図 7-13 マッチングスコアヒートマップ (PCLG マップ利用時)

リング付近で自己位置をロストした原因

- リング付近では、建設中のパビリオンが再現されておらず、またリングの 3D 都市モデルが検証用道路を境に整備されておらず、ガードレールが設置されていたため、特徴点とするには高さが足りなかった

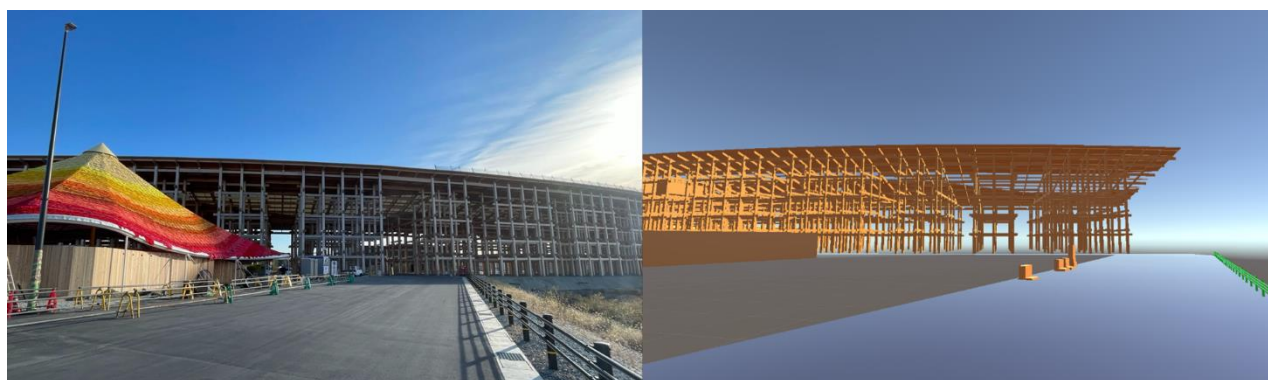


図 7-14 左：リング付近の様子（12 月 7 日撮影時）右：3D 都市モデル整備時点

ベース付近でスコアが低下した原因

- ベース付近では、停留所の構造物が 3D 都市モデルで再現されておらず、また右側のパビリオンが球状で点群マッチングに不向きな形状のため、マッチングスコアが低下した

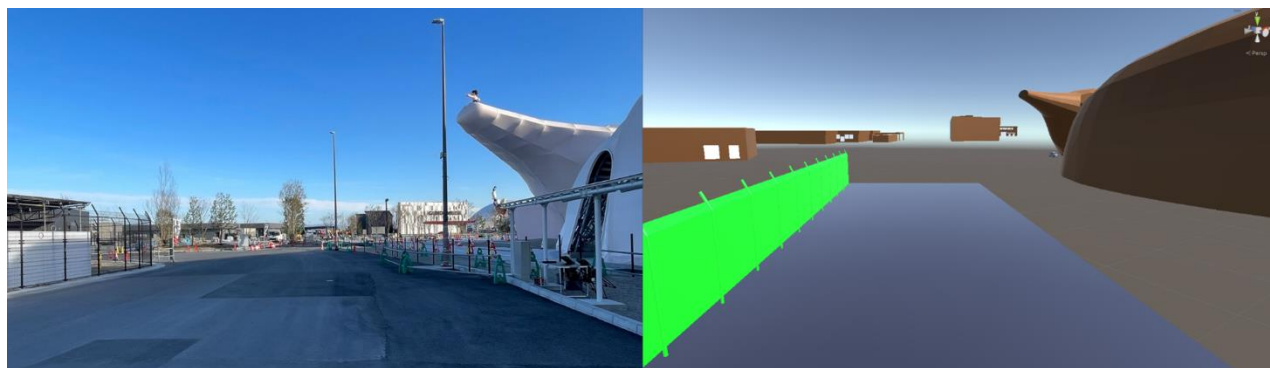


図 7-15 左：ベース付近の様子（12 月 7 日撮影時） 右：3D 都市モデル整備時点

[PCLG マップ+RTK 測位を用いた自己位置精度]

RTK 測位による自己位置推定の補完をすることで、全てのエリアでマッチングスコア 2.0 以上を達成し、自己位置をロストすることなく自動運行することができた。リング付近では、RTK-GNSS 測位による補完を実施しない場合では自己位置をロストしていたが、補完が有効に働いたことで自動運行を達成できた。

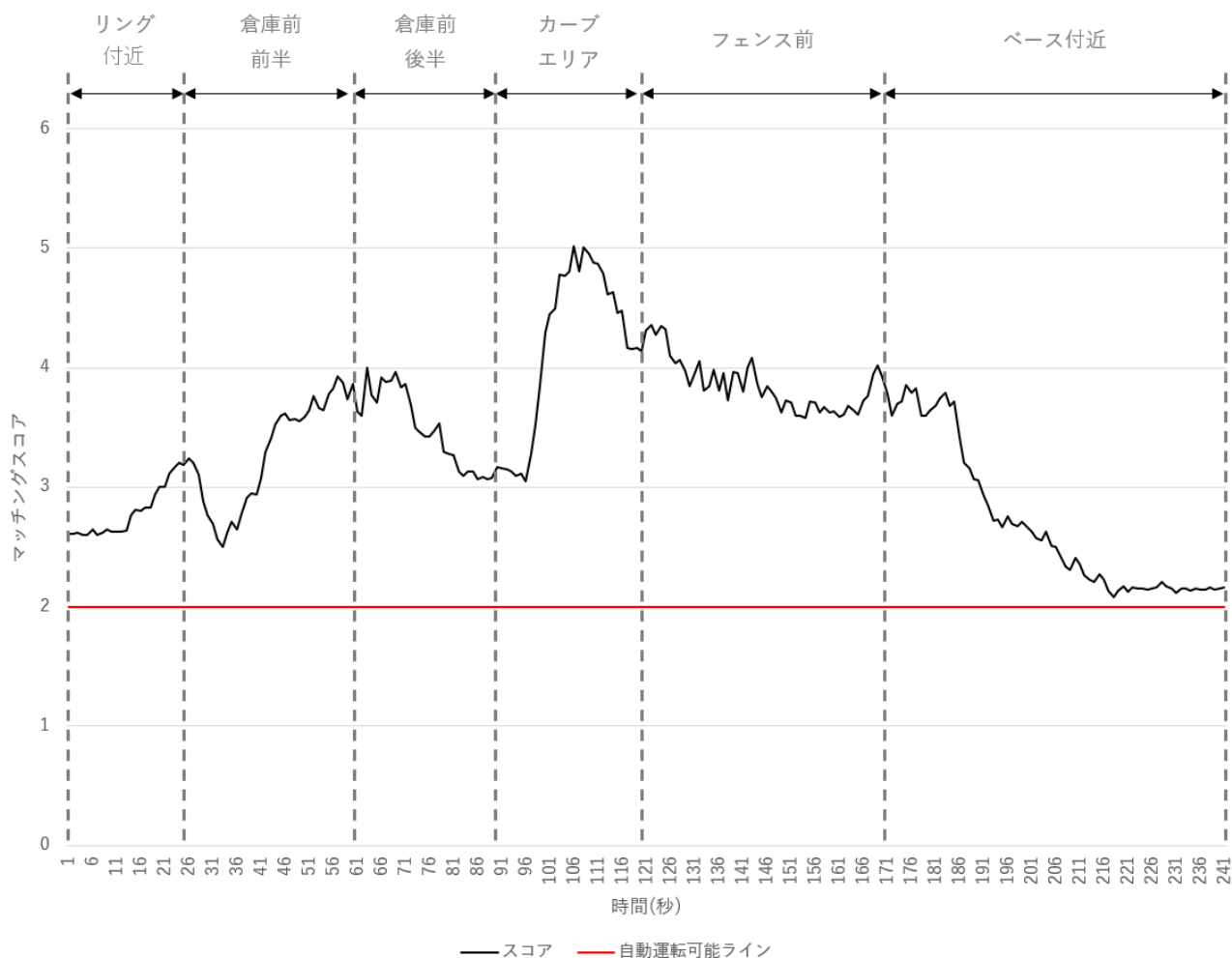


図 7-16 マッチングスコア推移グラフ (PCLG マップ+RTK 補正あり)



図 7-17 マッチングスコアヒートマップ（PCLG マップ+RTK 補正あり）

7-2. 水平・垂直方向の自動運航の検証

7-2-1. 検証目的

- ドローン・車両の連続運航により水平・垂直方向の輸送ができるかを確認し、KPI の達成可否を検証する

7-2-2. KPI

表 7-8 KPI 一覧

No.	評価指標・KPI	目標値	目標値の設定理由	検証方法サマリー
1	水平・垂直方向の自動運航を複数回試行し、自動運航を継続達成できるか	3 回以内で達成	● 水平・垂直方向の自動運航を継続達成できたか（達成/未達）を評価する	【前提条件】 ● 3D 都市モデル（LOD2~4） ● 天候：晴れ又は曇り ● 風速：5m 以下 ● その他： ➢ 車両に搭載するドローンの固定・固定解除：周囲の安全に配慮し人間が行う（自動運航の対象外） ➢ 輸送物：爆発物・危険物ではないもの ➢ その他の条件：工事現場内では、実証車両・ドローン周辺に他の車両が走っていない状況とする 大阪府大阪市夢洲地区において、ドローン・車両による水平・垂直方向の自律運航を行う
2	150g 程度の、本実証実験で使用するドローンに搭載可能な小型物品を載せた運航を達成できるか	達成	● 物流分野での実用化を見据え、本実証実験で使用するドローンに搭載可能な小型物品を積載した状態での自動運航を達成できたか（達成/未達）を評価する ● ドローンの可搬重量、搭載するセンサー類の積載重量から小型物品の重量（150g 程度）を算出	
3	SENSYN CORE Pilot からの走行開始指示が 10 秒以内に Autoware で受信できるか	10 秒以内	● システムの社会実装を見据え、SENSYN CORE Pilot からの指示時間が、自動運航を行う上で許容範囲であったか（達成/未達）を評価する	

7-2-3. 検証方法と検証シナリオ

- 以下の前提条件を満たす場合に検証を実施する。前提条件が満たせない場合は、時間又は日程をあらためる（前提条件が満たせない場合に備え、予備日を設けた）
 - 天候：晴れ又は曇り
 - 風速：5m 以下で行う
- 検証では、まず車両、次にドローンの順番で自動運航を行う
 - 地上に出発地点を、大阪・関西万博のシンボルである大屋根（リング）（以下、リング）上部の歩道（以下、リングスカイウォーク）にドローンが着陸できる到着地点（目的地）を設定
 - 車両の荷台にドローンを固定し、車両が出発地点からドローン離陸予定地点付近まで自動走行
 - 走行完了後、車両上からドローンが飛行を開始し、ドローンが目的地まで自律飛行を行うことで、水平・垂直方向の自動運航を実施
 - SENSYN CORE Pilot からの走行開始指示が数秒以内に Autoware で受信できるか確認する
- また、サブの検証項目として、以下も確認する
 - 150g 程度の、本実証実験で使用するドローン（DJI Matrice 300）に搭載可能な小型物品を載せて運航できるか

表 7-9 検証シナリオ一覧（水平・垂直方向の自律運航の継続）

No.	検証方法	エリア	検証のポイント
1	車両・ドローンによる 自律飛行	大阪府大阪市 夢洲地区	水平・垂直方向の自動運航を複数回試行し、自動運航を継続達成できるか
2			150g 程度の、本実証実験で使用するドローンに搭載可能な小型物品を載せて運航できるか
3			SENSYN CORE Pilot からの走行開始指示が数秒以内に Autoware で受信できるか

7-2-4. 検証結果

LiDAR 自己位置推定の事前地図として 3D 都市モデルを活用した車両・ドローンの連続自動運航を複数回実施し、2 回目で達成した。

また、車両・ドローン自動運航実施時に、水平・垂直方向輸送のためドローンに小型物品（目的地であるリングスカイウォークの花壇に植えられている花と同じ品種の種と固形肥料、合計約 170g）を載せ、目的地へ無事に物品を届けることできた。

さらに、車両の自動走行開始時に SENSYN CORE Pilot からの走行開始指示について、Autoware で受信するまでに掛かった時間は、実施したどの自動運航においても 3.0 秒未満であった。

表 7-10 検証結果サマリー

赤セル：KPI 達成

青セル：KPI 未達

検証内容	評価指標・KPI	目標値	結果	示唆
			評価	
車両・ドローンによる自律飛行	水平・垂直方向の自動運航を複数回試行し、自動運航を継続達成できるか	3 回以内で達成	2 回目で達成	● 実証場所の特性上、3D 都市モデルの整備時点（2024 年度）では存在しなかった建築物や資材等が実証時に存在した。その結果、3D 都市モデルと実際の構造物との間に差が生じ、点群マッチングの精度が下がる場面があったが、前章 7-1 で詳述した通り、RTK 測位データを活用することで自己位置をロストせずに自動運航を継続達成できた。 ● GNSS ユニットがない無人機でも 3D 都市モデルを活用して安全に自動運航する場合、例えば 3D 都市モデルと実際の構造物との間に差異がある場合に点群マップを更新する機能などがあれば、よりロバストで安全な自動運航が実現できると考えられる。
	150g 程度の、本実証実験で使用するドローンに搭載可能な小型物品を載せて運航できるか	達成	達成	
	SENSYN CORE Pilot からの走行開始指示が 10 秒以内に Autoware で受信できるか	10 秒以内	3 秒	

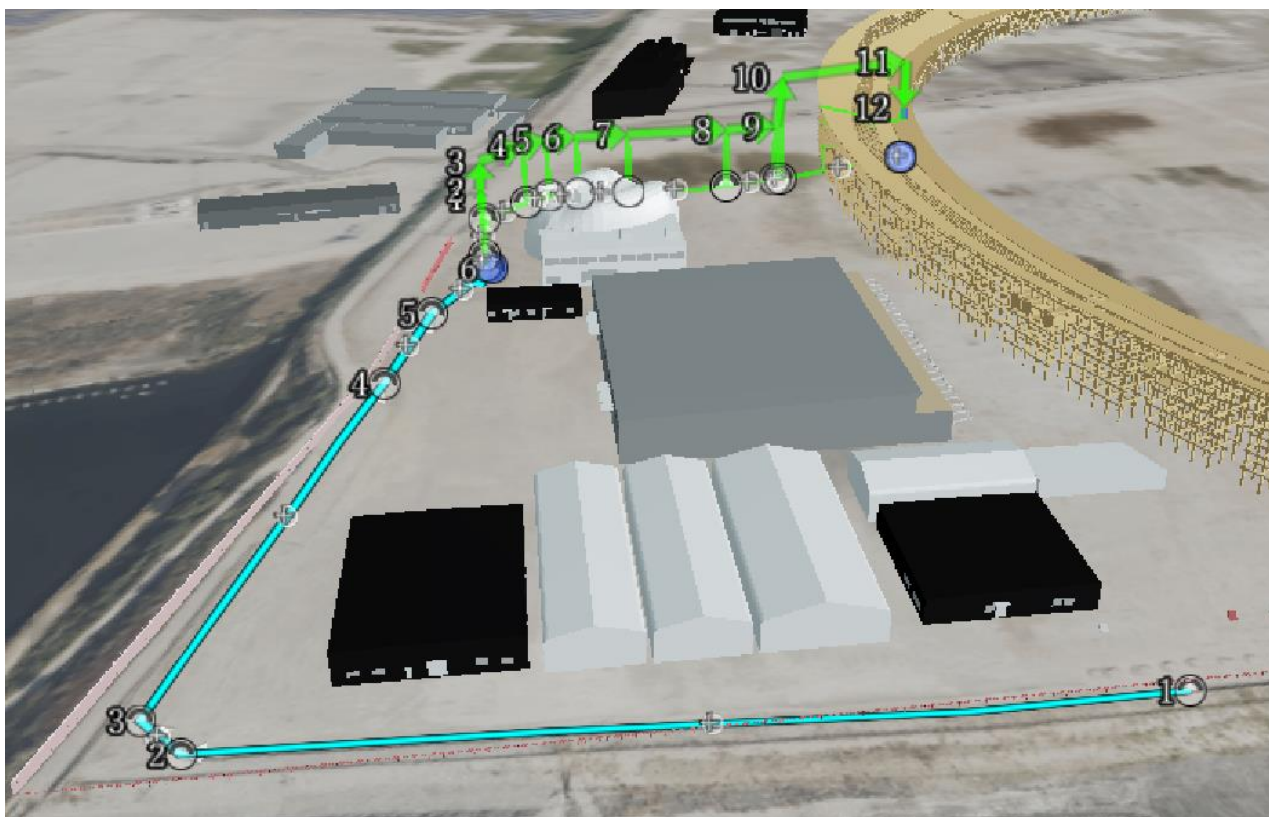


図 7-18 3D 都市モデルを反映した SENSYN CORE Pilot を用いた、
車両・ドローン統合ルート作成（実証実験時に用いたルート）
水色の線：車両ルート、緑色の線：ドローンルート



図 7-19 車両・ドローン統合ルートのアップロード画面

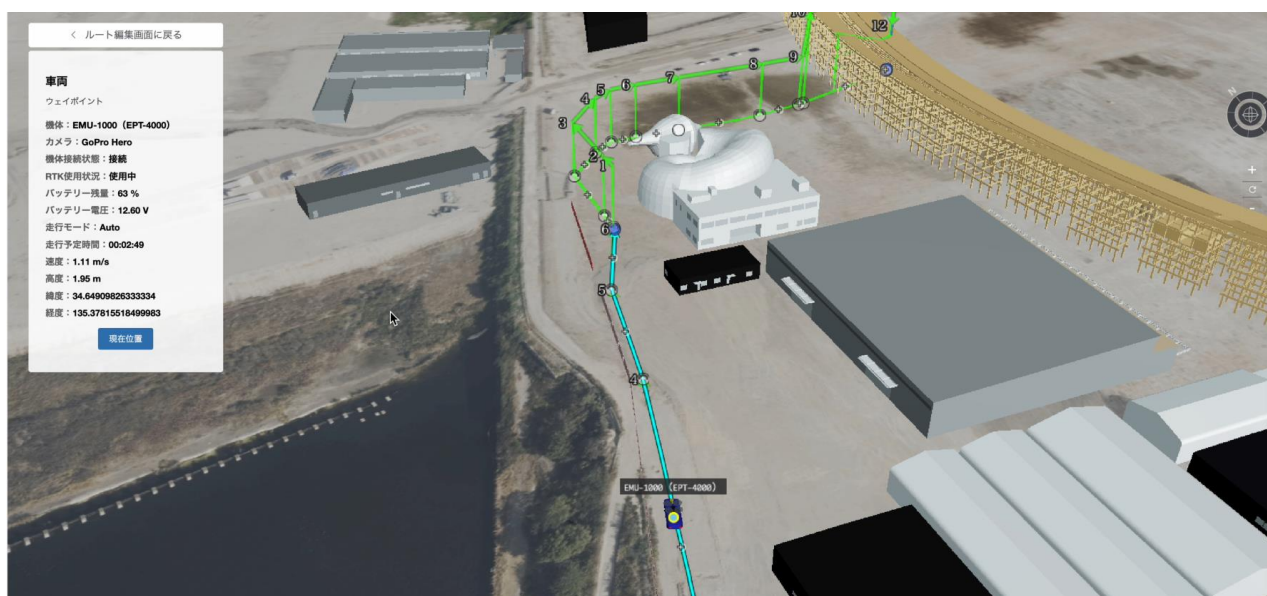


図 7-20 車両自動走行中、遠隔モニタリングをしている様子

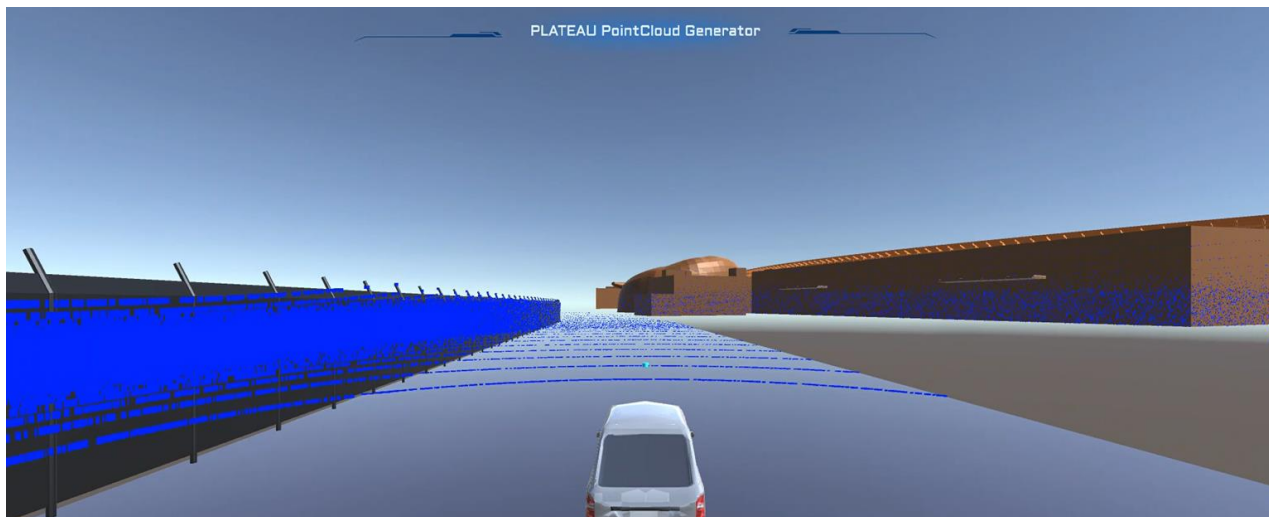


図 7-21 図 7-3 時の 3D 都市モデルを用いた点群マップ作成の様子

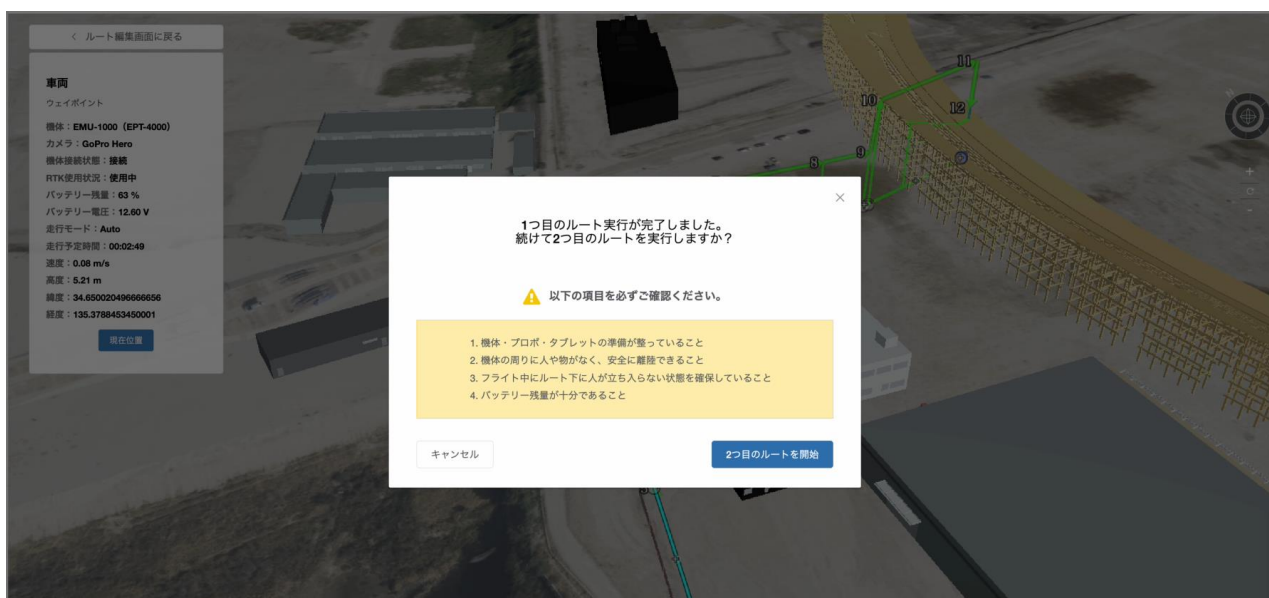


図 7-22 車両の自動走行完了後、シームレスにドローンの自律飛行を開始

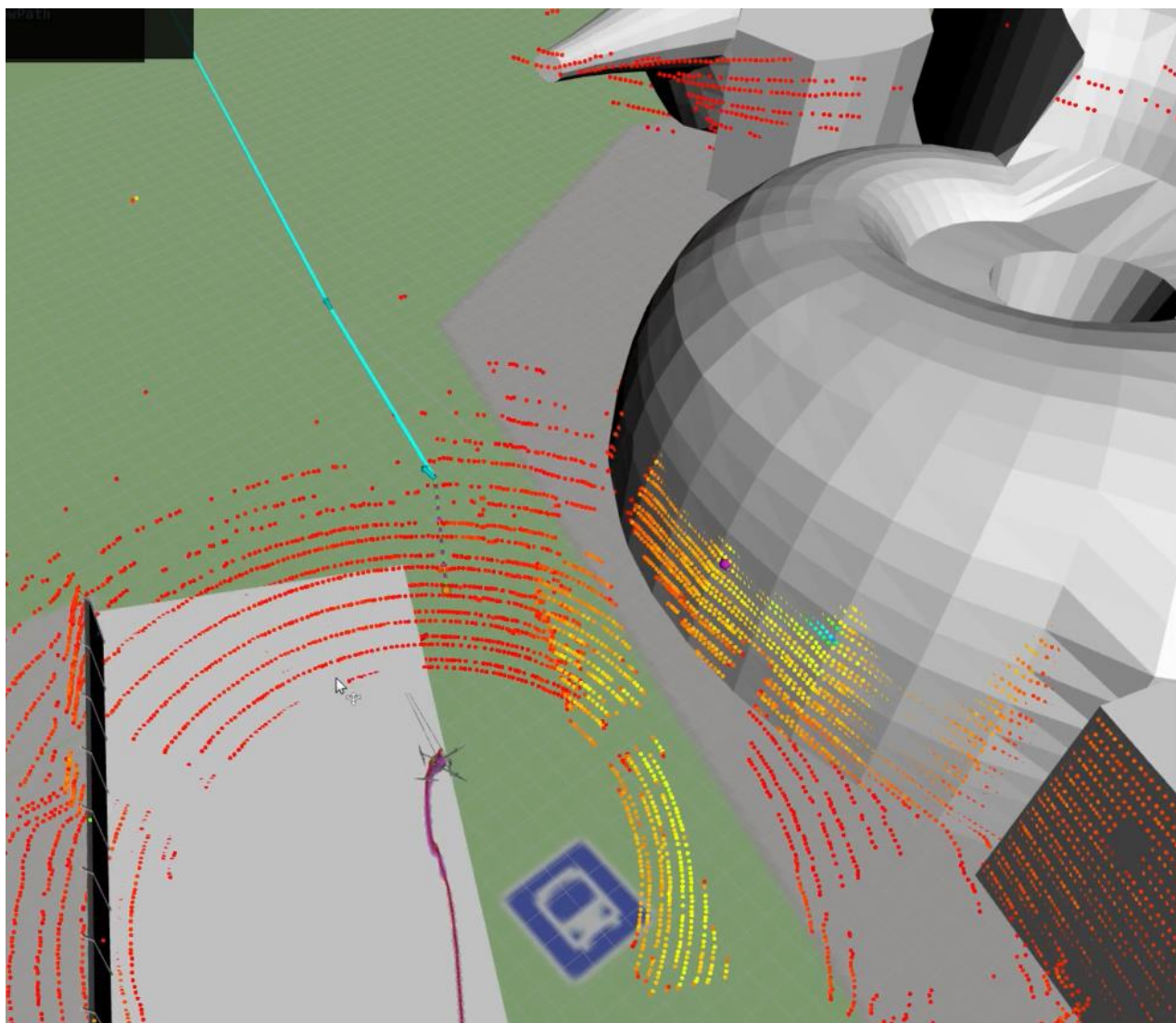


図 7-23 3D 都市モデルを利用して LiDAR 自己位置推定を行っている様子①（パビリオン脇）

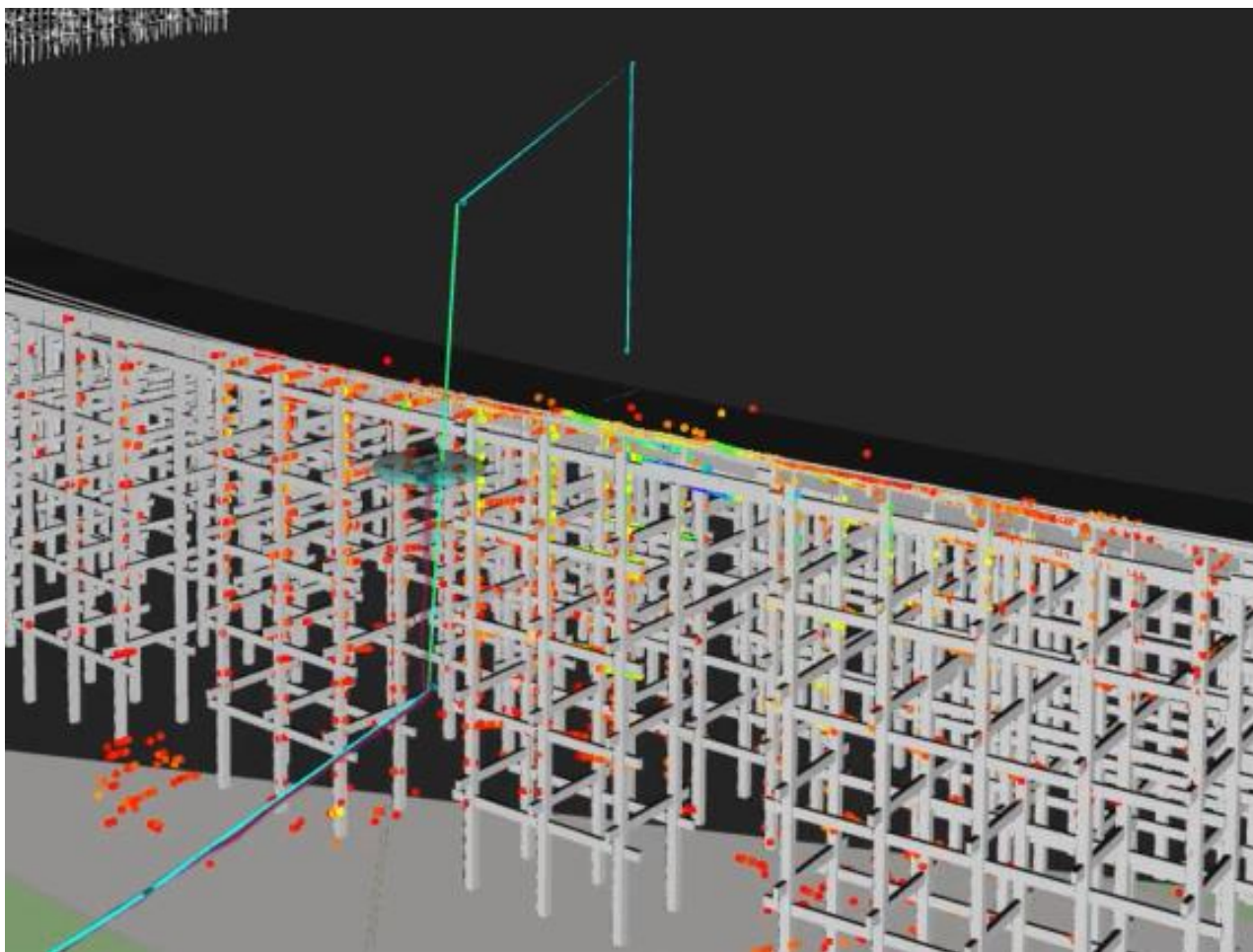


図 7-24 3D 都市モデルを利用して LiDAR 自己位置推定を行っている様子②（リング付近）



図 7-25 評価指標・KPI[サブ①]達成のため、ドローンに搭載した輸送物

目的地であるリングスカイウォークの花壇に植えられている花と同じ品種の種と固形肥料、合計約
170g



図 7-26 評価指標・KPI[メイン]（水平・垂直方向の自動運航）及び評価指標・KPI（小型物品の輸送）を達成

8. 実証技術の非機能要件の検証

8-1. 検証目的

- 実証実験を実施するために必要な時間、安定してシステムが稼働することを検証する
- 実証実験を安全に実施するために必要なセキュリティが担保されることを検証する
- ユーザーが使いやすいシステムであることを検証する

8-2. KPI

表 8-1 非機能要件の KPI 一覧

カテゴリ	ID	項目	詳細
可用性	NR001	システムの連続稼働時間	● メンテナンス時間等を除き、原則として 24 時間 365 日連続稼働すること
	NR002	安定動作時間	● 実証環境のため、1 時間以上安定動作すること
	NR003	天候対応	● 車両、ドローンともに晴天、曇天に対応できるよう対策を行うこと ● ドローンについては、風速 5m 毎秒まで対応できるよう対策を行うこと
性能・拡張性	NR004	ドローンに搭載する Edge PC	● ドローンに搭載するため、小型かつ軽量であること ● Edge PC 内で点群処理などを行う必要があり GPU が必要となることから、グラフィック処理が行えるプロセッサが搭載されていること
	NR005	画面操作性能	● 一般的に操作しやすい環境にするために、UI で視覚的に分かりやすい状態で入力設定が可能であること
運用・保守性	NR006	セキュリティ	● 対象となるデータ及びシステムの保守のために、通信に暗号化を行うこと ● 必要最低限のサービスのみアクセス可能となるよう、ルーター、ファイアウォール等でパケットフィルタリングすること
	NR007	認証	● 利用ユーザーを特定するためにユーザー名、メールアドレスを設定したアカウントを発

			行する
--	--	--	-----

8-2-1. 検証方法と検証シナリオ

表 8-2 非機能要件の検証方法

対象項目	品質評価項目	目標値	期間の単位	アクティビティ
SENSYN CORE Pilot	安定動作時間	● 1 時間	2024/10～12 月	● 運用テストによる検証
SENSYN CORE Pilot	画面操作性能	● 選択肢は「とても不満」を 1、「とても満足」を 5 とした 5 段階で設定。 ● 回答を集計し、各選択肢の選択率から評価（各設問で、過半数の 4 以上の回答を目標とする）	2024/10～12 月	● ヒアリング・アンケートによる検証

8-2-2. 検証結果

実証実験を実施するにあたり、安定動作時間、画面操作性能について、目標値を達成できた。

表 8-3 検証結果サマリー

赤セル：KPI 達成

青セル：KPI 未達

検証内容	評価指標・KPI	目標値	結果	示唆
可用性	安定動作時間	1 時間	8 時間以上	● 目標値を超える時間安定稼働した
性能・拡張性	画面操作性能	● 選択肢は「とても不満」を 1、「とても満足」を 5 とした 5 段階で設定 ● 回答を集計し、各選択肢の選択率から評価（各設問で、過半数の 4 以上の回答を目標とする）	達成	● 次章で記載する

9. BtoB ビジネスでの有用性検証

9-1. 検証目的

実証仮説に基づき、以下の検証目的を設定する。

【検証目的】

3D 都市モデルを活用した車両・ドローンの運行管理/モニタリング業務について、昨年度実証実験で改修を加えた SENSYN CORE Pilot に追加開発を加え車両・ドローンを同時に管理するインタフェースを備えることで、専門的な知識を持たないユーザーでも簡単に利用可能となることを示す。

主に以下の 2 点について、BtoB ビジネスに向けた有用性検証を行う。

- システムのユーザビリティ検証
 - 今年度開発した本システムにおける UI/UX の分かりやすさ・使いやすさを確認する
- 既存システムとの利便性比較
 - 車両・ドローンの経路計画について、以下の方法を比較し、利便性や操作性の違いを確認する
 1. 昨年度までと同様に、車両・ドローン別々のシステムを用いる方法
 2. 車両・ドローンで統合されたシステム（以下、新規開発システム）を用いる方法
 - 今回の実証実験で実際に自動運航を行った本番ルート（大阪市）（以下、本番ルート）のデータを使用し、昨年度までの実証実験で改修済みの車両・ドローンで独立したシステム（以下、既存システム）と本年度追加開発した車両・ドローンの統合システムの両方で同様のシミュレーションを行うことで、利便性や操作性の違いを確認する

9-2. 検証方法

検証方法としては、被験者に対してデモンストレーションを採り入れたヒアリング・アンケートを実施する。
（ヒアリング・アンケートの項目については「9-4.ヒアリング・アンケートの詳細」で記載）

事業者向けヒアリングの実施方法

（被験者の所属部署ごとに個別にヒアリング会を実施するが、内容は同様）

- 会場：自社の会議室など
- 機材：体験・デモンストレーション用に以下のスペックの社用 PC を用意する。
 - CPU：12th Gen Intel® Core™ i7 以上
 - メモリ：16GB 以上
 - OS：Windows 10 以降、Linux 18.04
 - 通信環境：各社で用意された社用 Wi-Fi
- ヒアリング・アンケートに利用するルート
 - 本番ルートと同様のエリア（対象物・City GML などを含む）において、被験者が自ら車両とドローンのルートを作成する。

9-3. 被験者

本ユースケースでは、既にドローンを自律飛行で使用している事業者（主に建設関連の事業者）や自動搬送車両の自動運転の実証実験を行っている事業者のうち、飛行・走行ルートの計画、又は実際に飛行・走行のオペレーションを行う部署の担当者をターゲットとした。

本実証実験では、これらのユーザーに該当する以下の方々にヒアリング・アンケートを行い、本システムの価値を検証した。

表 9-1 被験者リスト

分類	具体名称	部署	役職	担当業務	人数
事業者/ ユーザー	大手建機レンタル メーカー	道路機械事業部 ICT 施工推進課	課長	ドローン飛行計画 ソフトウェア運用	1 名
	無人ロボット運用 代行事業者	モビリティ事業部	オペレーター	無人ロボットの運航計 画、ソフトウェア運用	2 名
	道路測量会社	モビリティ事業部	マネージャー	自動走行車両の走行計 画、ソフトウェア運用	1 名
	センシンロボティ クス	ソリューショングル ープ	マネージャー	ドローン飛行計画、 ソフトウェア運用	1 名
			オペレーター	ドローン飛行オペレーシ ョン、ソフトウェア運用	1 名

9-4. ヒアリング・アンケートの詳細

9-4-1. アジェンダ・タイムテーブル

表 9-2 アジェンダ・タイムテーブル

No.	アジェンダ	所要時間
1	実験の目的を説明	10 分
2	作業手順のデモンストレーション、説明（既存と新規開発）	20 分
3	操作の体験①（既存システム）	20 分
4	操作の体験②（新規開発システム・本年度ユースケースで開発したシステム）	20 分
5	操作感のヒアリング	20 分
6	アンケート回答（Google フォームでの回答）	10 分

9-4-2. アジェンダの詳細

表 9-3 アジェンダの詳細

No.	アジェンダ（再掲）	内容
1	実験の目的を説明	● 本実証実験でアプローチする課題や背景の説明 ● 本実証実験の比較対象となる従来手法の説明 ● 本実証実験で用いるシステムの提供価値 ● システムの全体像の説明
2	作業手順のデモンストレーション、説明 （既存と新規開発）	● 既存システム ➢ 車両について、PC 上で Autoware を利用した車両経路計画（車両ルートの作成）機能のデモンストレーション、操作説明 ➢ ドローンについては、SENSYN CORE Pilot 上で、昨年度ユースケースで開発したシステムを用いた CityGML 及び BIM を利用したドローン経路計画（ドローンルートの作成）機能のデモンストレーション、操作説明 ● 新規開発システム ➢ SENSYN CORE Pilot 上で、本年度ユースケースで開発したシステムを用いた CityGML 及び BIM を利用した車両・ドローンの統合経路計画機能のデモンストレーション、操作説明 ● SENSYN CORE Pilot 上での自動運航ルート作成 ➢ 被験者者は SENSYN CORE Pilot 上に車両とドローンのルートを作成 ➢ そのルート上を、仮想環境下のテレメトリーデータを基に、車両・ドローンのアイコンが移動することを確認 ➢ 本番ルートでの自動運航を想定

3	操作の体験①	● 上記デモンストレーション内容を事業者が体験 ● 既存システムにて経路計画～自動運航開始ボタンクリックまでの総作業時間を計測
4	操作の体験②	● 新規開発システムにて経路計画～自動運航開始ボタンクリックまでの総作業時間を計測
5	操作感のヒアリング	● 質疑応答及びヒアリングを実施
6	アンケート回答	● Google フォームでアンケート回答を依頼し、その場で回答（10分）、送信を依頼

9-4-3. 検証項目と評価方法

既存システムとの比較とユーザビリティ評価を検証項目とし、それぞれ定量・定性的に評価した。

表 9-4 検証項目と評価方法

検証観点	No.	検証項目	定量評価	定性評価
既存のシステムと利便性比較	1	総作業時間	● ストップウォッチで経路計画～自動運航開始ボタンクリックまでの作業時間を計測 ● それぞれの作業時間を比較	-
	2	ルート作成操作が直感的か	● 対象ユーザーに既存のシステム及び本年度開発したシステムを体験していただいた後、アンケートを実施 ● 選択肢は「大幅に悪化した」を1、「大幅に改善した」を5とした5段階で設定。 回答を集計し、各選択肢の選択率から評価（各設問で、過半数の4以上の回答を目標とする）	● アンケートの各設問に自由記入欄を設定
	3	モニタリング操作が直感的か		
	4	画面表示にストレスがないか		

9-4-4. 実証実験の様子

会議室で本システムの操作手順を説明している様子



図 9-1 操作手順の説明

被験者が Autoware を利用している様子

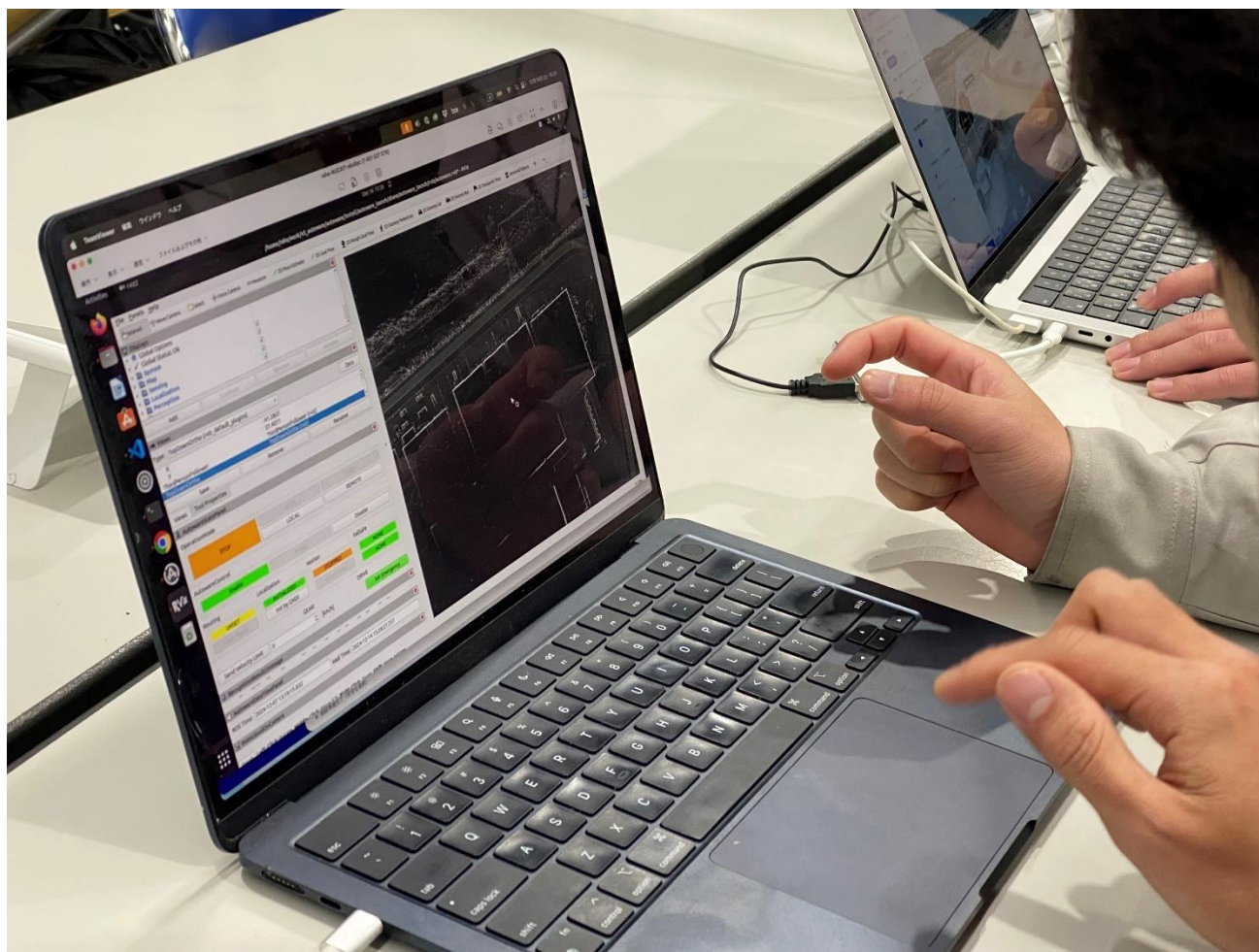


図 9-2 被験者による Autoware 体験

被験者が SENSYN CORE Pilot で自動運転ルートを作成している様子

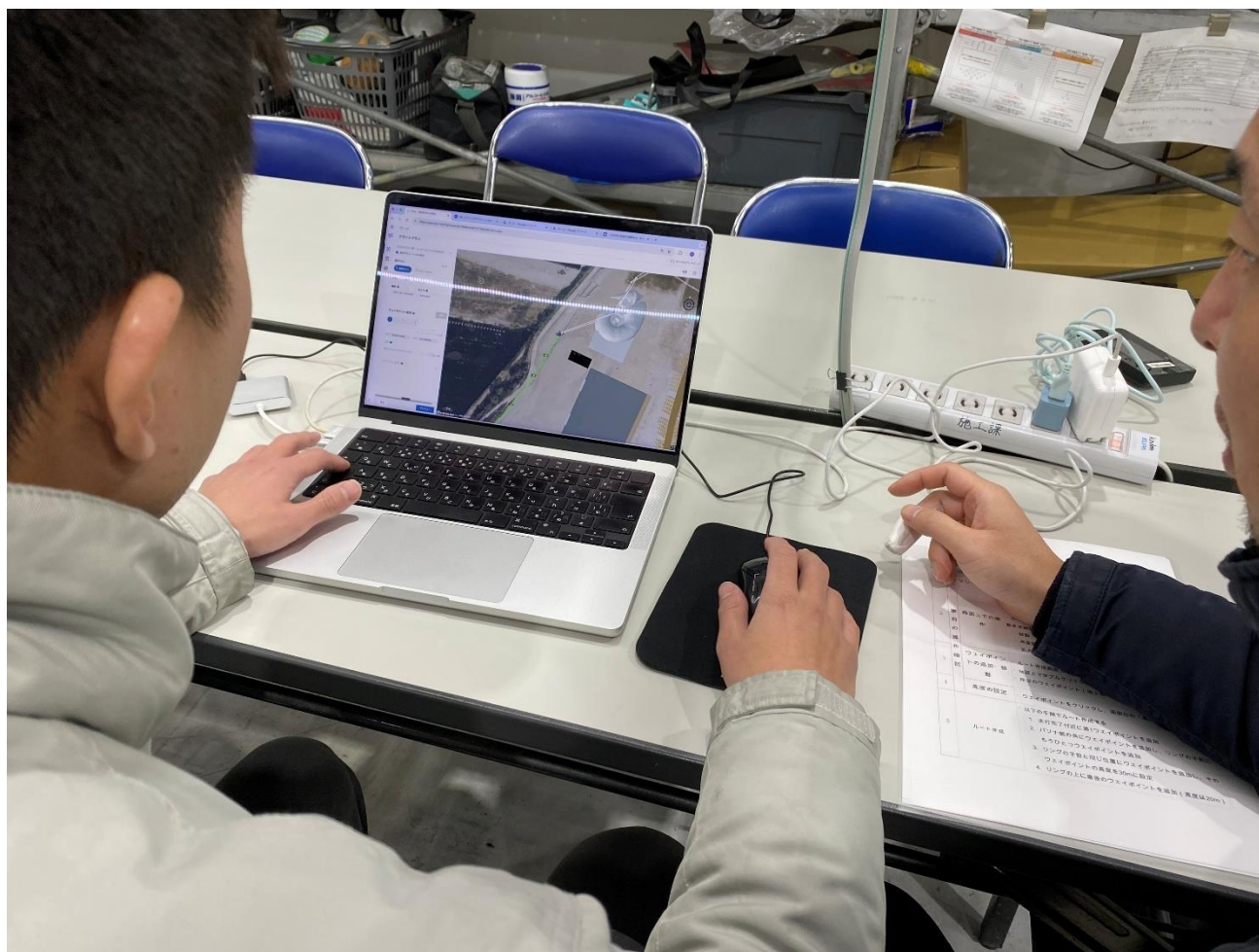


図 9-3 被験者による SENSYN CORE Pilot でのルート作成

9-5. 検証結果

既にドローンを自律飛行で使用している事業者（主に建設関連の事業者）や自動運転の実証実験を行っている事業者のうち、飛行・走行ルートの計画、又は実際に飛行・走行のオペレーションを行う部署の担当者を対象に検証を行った。担当業務の内訳は、ルート作成担当が 83.3%、自動運航オペレーション担当が 33.3%であった。（合計が 100%を上回るが、これは担当者の中には両方の業務を行うケースがあり、重複回答を可としたため）

その結果、全ての検証項目において KPI で設定した目標以上の定量評価を得ることができ、実業務を想定し必要十分な機能及びインタフェースを兼ね備えたシステムを提供できたことがわかった。また、定性評価においても、一部課題はあるものの好意的なコメントを得た。

今回開発したインタフェースは「ルート作成」、「車両/ドローンルートの統合」、「自動運航ルートのアップロード」、「自動運航開始指示」、「車両/ドローンの位置表示」の 5 画面で制御を行う仕様となっている。

その中でも特に「ルート作成」については、従来オペレーターが課題と感じていた「（Autoware は）動作が重い」、「操作が難しい」、「車両、ドローン別々にルートを作成しないといけなため、3D 都市モデルを利用した一連のルートがイメージしにくい」ことについて、直感的に触れる SENSYN CORE Pilot に操作が一元化されたことで動作が軽く、また操作についても「説明を受けずとも触れるようになった」（操作が簡易になった）といったフィードバックを得ており、価値を提供することができたといえる。

ユーザビリティに関しては、これまで車両の自動走行については Autoware を使用したシステムとなっていたため、システムへの理解及び知見がないと操作が困難な仕様になっていた。そのため今年度は 3D 都市モデルを活用した車両/ドローンを統合した形の経路計画について、自動搬送車両やドローン関連の技術者ではなくてもオペレーション可能な GUI を含むシステムを実現し、UI/UX 面でユーザビリティの高さについて一定評価を得ることができた。

また、SENSYN CORE Pilot のマップ上に表示される 3D 都市モデルについて、オープンデータの建築物モデル（3D Tiles）表示にのみ対応していたが、業務を実施する上では車両が出発ないし帰還する際に使う格納庫や基地のような場所の情報が明示されていることが望ましく、そのような仕様への対応及び機能拡張が必要であることなど、より簡便な操作を実現するための課題点が明確になった。

1) 既存のシステムとの利便性比較

● 作業時間の比較

ストップウォッチで経路計画～自動運航開始ボタンクリックまでの作業時間を計測。それぞれの作業時間を比較した。

新規開発システムについて、全参加者（6/6 名）が既存システムと比較し半分程度の時間で作業を完了した。なお、SENSYN CORE Pilot での作業時間について、ドローン部分については新規開発システムにおいても既

存システムと同じものを継続利用しているため、これら操作時間の差分としては、そのほとんどが「車両の自動走行における Autoware と SENSYN CORE Pilot の作業時間の差」と言える。

このことから、運用性の向上が明らかになったと言える。

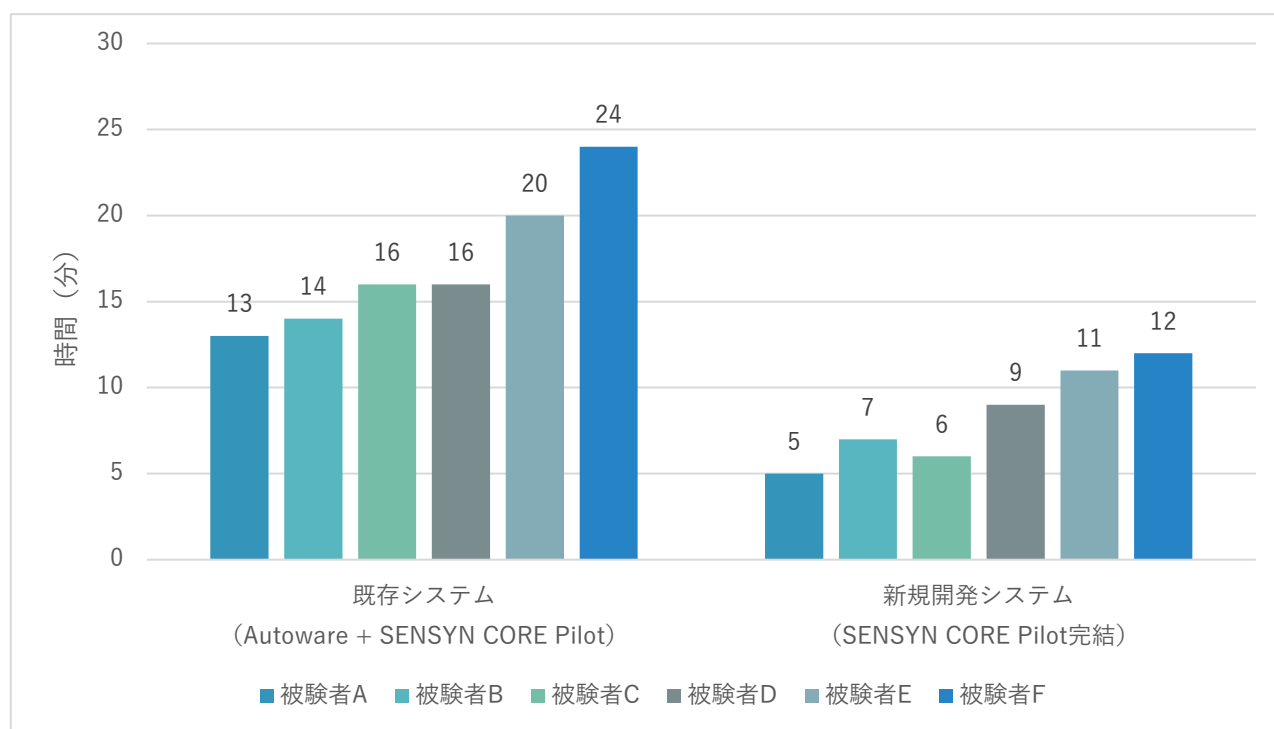


図 9-4 作業時間の比較

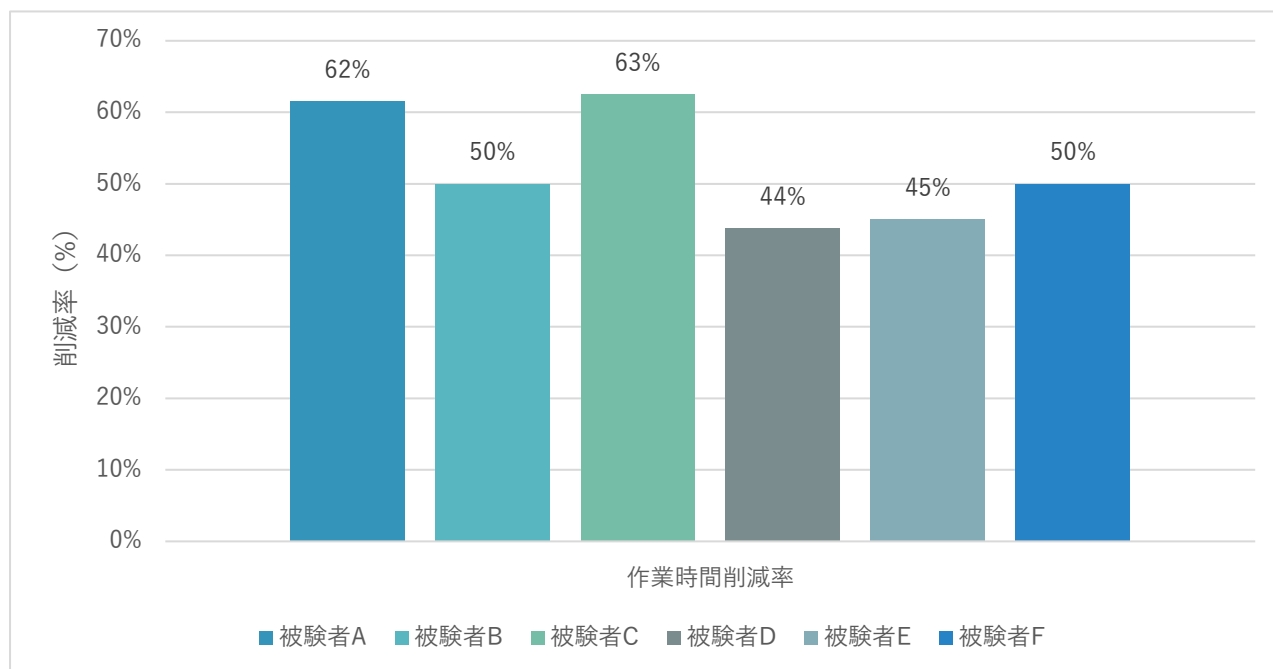


図 9-5 作業時間の削減率

なお、今回計測対象から外したソフトウェア立ち上げ作業について、計測対象から外した理由として、Autoware は立ち上げるのに Ubuntu やコマンドラインの知識が必要なためエンジニアでないと操作が困難で

あることが事前に予想されたことが挙げられる。そのため、本検証内では Autoware 立ち上げ作業を検証実施者（株式会社アダウェアジャパン）が行った。

こちらについて、本ヒアリング内でソフトウェア立ち上げ作業を見学した参加者から「Autoware をターミナルで立ち上げるのはハードルが高い」、「(Autoware をそのまま) エンドユーザーが利用するのは無理だと思う」というフィードバックが実際に寄せられた。このことから事前予想が正しかったことが裏付けられ、また同時に、立ち上げが容易な Web アプリである SENSYN CORE Pilot で車両側の操作が可能になった、新規開発システムの優位性が定性的に示されたと言える。

計 6 名の回答者のうち、「大幅に改善した」もしくは「改善した」を選択した割合が、ルート作成操作の分かりやすさについては 100%、モニタリング操作の分かりやすさについては 83%、画面表示の快適性については 83%であった。

全体を通して UI の分かりやすさは高く評価されており、技術リテラシーを問わず活用可能なシステムとして受け止められていることがわかる。

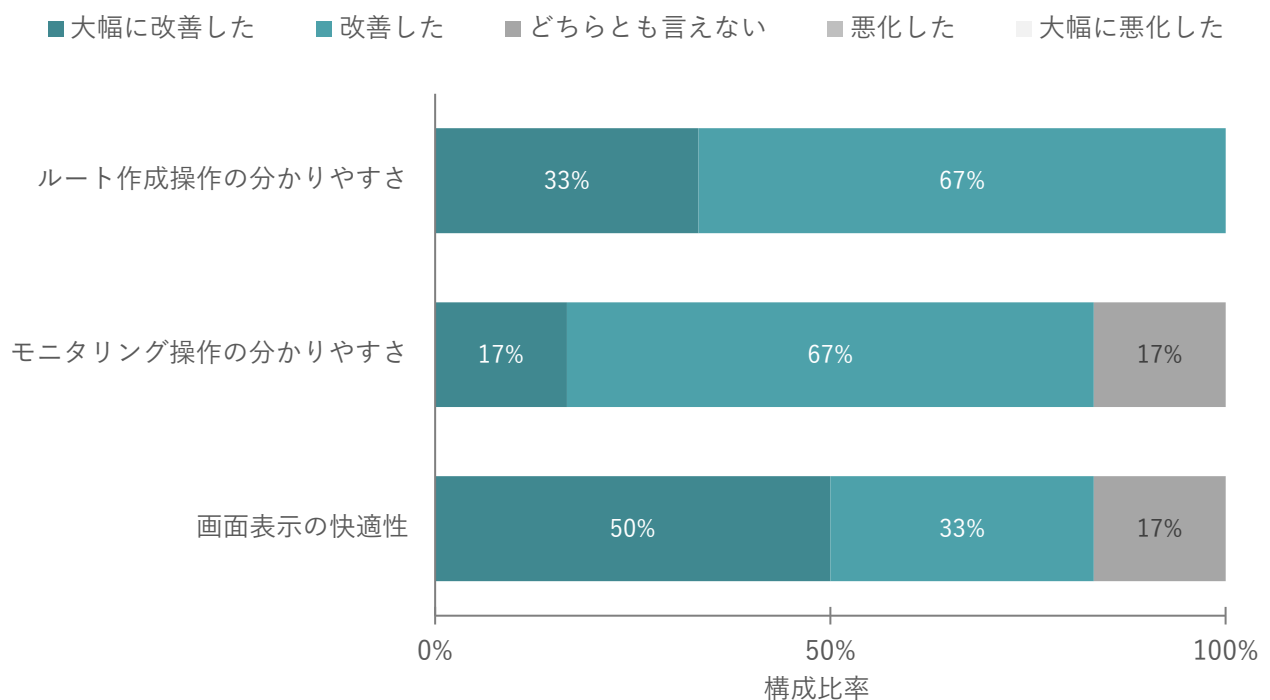


図 9-6 モニタリング操作が直感的か（走行・飛行中の状況が分かりやすいか）

表 9-5 関連する定性コメント

No.	検証項目	関連する定性コメント
1	ルート作成操作の分かりやすさ	【大幅に改善した】 ● 直感的に触れる SENSYN CORE Pilot に一元化されたことで説明を受けずとも触れるようになった。真上表示、真横表示などがあるとより経路が設定しやすいと感じた ● 動作が軽く、操作感が良い ● Autoware をターミナルで立ち上げるのはハードルが高い ● Autoware をそのままエンドユーザーが利用するのは無理だと思う 【改善した】 ● 往路の情報を基に、復路を自動で作れると良いと思う ● 車両の初期位置について、実際は格納庫・基地のような場所から発信すると思うので、その情報があるとルートが作りやすい
2	モニタリング操作の分かりやすさ	【大幅に改善した】 ● 実際に自動運航している場所を画面で見られるのは非常にいいと思った 【改善した】 ● UI（マップの視点/拡大縮小など）が変更しやすく分かりやすい 【どちらとも言えない】 ● 走行中のカメラ映像はモニタリング画面で一緒に見られると良いと思う
3	画面表示の快適性	● 定性コメントなし

10. 成果と課題

10-1. 本実証で得られた成果

10-1-1. 3D 都市モデルの技術面での優位性

実証実験を通じて、以下のような 3D 都市モデルの技術面での優位性が示された。

表 10-1 3D 都市モデルの技術面での優位性

大項目	中項目	小項目	3D 都市モデルの技術面での優位性
システム全体	システム・機能	複数視点のシームレスな切替え	● 従来は視点切り換えができなかったが、3D 都市モデルの詳細な地形データと建築物データを利用することで、ユーザーは高精度な俯瞰視点と 1 人称視点の切り替えが可能となり容易にルート作成ができる ● Cesium に元々表示されている 2D 地図データを使ったルートの作成に加え、3D 都市モデルに含まれる建物や道路の立体データを利用することで構造物を考慮したルートの作成ができる
		LOD4 利用による 3D オブジェクトの位置合わせ	● 従来の BIM モデルを地図上に表示する前に位置合わせが必要だったが、BIM を LOD4 に変換した位置情報を持つ 3D 都市モデルを用いることで、地図上での表示が容易にできる
車両	システム・機能	現場検証の工数減	● 従来は現場にて点群取得し 3D マップを作成する必要があったが、3D 都市モデルの形状を使うことで、点群マップの生成を PCLG 上で行うことができ、現場に行く前に走行ルートを設計することができる
		高さのある 3D 都市モデルの有用性	● 一定以上の高さのある 3D 都市モデル（フェンスなど）を用いることで、点群マッチングの精度が向上できる
ドローン	その他	ドローン用点群マップ作成の簡便化	● ドローン用点群マップを作成する場合、SLAM による地図作成技術のコストの高さや精度の不足などの課題があったが、オープンデータである 3D 都市モデルを利用することで事前のデータ取得や取得データを基にした地図作成の必要がなく、低コスト・高精度な点群マップを簡便に作成できる

10-1-2. 3D 都市モデルのビジネス面での優位性

表 10-2 3D 都市モデルのビジネス面での優位性

大項目	中項目	小項目	3D 都市モデルのビジネス面での優位性
システム全体	サービス開発 期間・コスト の削減	オープンデータによる開発・運用コスト削減	● 3D 都市モデルは公的なオープンデータとして整備されていることから、ドローン事業者としては追加コストや許可手続きを経ることなく短期間で開発・市場導入ができる
		整備範囲の広さによるビジネスの拡張性	● 3D 都市モデルの全国的な整備が進むことで特定の地域に限らず全国にビジネス展開できる
車両	サービス開発 期間・コスト の削減	事前調査費用の削減	● 3D 都市モデルが整備されていれば、現地へ行かなくても自動運転の精度が確認でき、本格的な検証を行う事前調査の費用を抑えることができる
ドローン	サービスの提供価値向上	LiDAR 用点群マップ作成の低コスト化及び汎用性の担保	● 各事業体でドローン用点群マップを作成する場合、SLAM による地図作成技術のコストの高さや精度の不足などの課題がある。一方で、3D 都市モデルを利用することで事前のデータ取得や取得データを基にした地図作成の必要がなく、低コスト・高精度な点群マップを簡便に作成できる ● 3D 都市モデルの利用で汎用性を担保できることで、より幅広いユーザーにリーチできる
		シミュレーションの高度化	● 3D 都市モデルを活用することで飛行ルートシミュレーションが高度化し、従来の手法では困難であった建物が密集する都市部や高層ビル間の正確な飛行ルートシミュレーションを実現、新しい物流サービスの展開ができる
		ユーザー体験価値の向上	● 3D 都市モデルによる立体的でリアルな都市空間の再現ができる ● 2D 地図データを使ったドローンルートの作成に加え、3D 都市モデルを活用して構造物を考慮したルートの作成ができる ● これにより、ユーザーによる GCS アプリの体験価値向上を見込むことができる
	サービス開発 期間・コスト の削減	ドローン用点群マップ作成工数の短縮	● ドローンでの屋内外をシームレスに移動する自律飛行のためには事前に手動飛行による点群マップの作成が必要不可欠であり、飛行のハードルは高かったが、3D 都市モデルの導入により、事前に点群マップ作成のた

			めのデータ取得を行わずに自律飛行用の点群マップが作成できるようになり、従来よりも工数を削減できる
--	--	--	--

10-2. 実証実験で得られた課題と対応策

表 10-3 実証実験で得られた課題と対応策

大項目	中項目	小項目	実証実験で得られた課題	課題に対する対応策
システム全体	システム (UI/UX)	ルート編集 (ウェイポイントの設定)	● ドローンの飛行計画時に緊急輸送道路を避けるための情報がマップ上に表示されていないため、安全な飛行計画の立案が困難	● 緊急輸送道路の情報をマップに表示する機能を追加し、飛行計画に反映する
			● 車両が出発や帰還する際に格納庫や基地の情報が明示されていないため、車両の管理が困難	● 格納庫や基地の位置情報を 3D 都市モデルに追加し、車両の管理を容易にする
車両	システム (機能)	特殊な形状の建築物に対する点群マッチングの精度	● 球状の建物の場合、点群マッチング精度が下がるため、正確な自己位置推定が困難	● 球状の建物だけでは精度が下がってしまうため、対面の別の建物でも点群マッチングを行えるようなコース設計を実施する
		建築中の建物に対する点群マッチングの精度	● 建設途中の建物が完成した後に、マッチング精度が下がるため自己位置推定が困難	● 大きな変化が発生した箇所のモデルを容易に取得・登録できる手法を開発する ● 大きな変化をノイズとして認識し、マッチングの対象外とする機能を開発する
		ガードレールなどの、高さが低い地物への対応	● ガードレールでは特徴点にするには高さが足りず、自己位置の精度が低いため、自己位置推定が困難	● ガードレールなどの地物に対応するため、車両低い位置に LiDAR を取り付ける
	システム (UI/UX)	SENSYN CORE Pilot でのルートアップロードエラー対応	● ユーザーの操作ミスによるものではあったが、車両情報のテレメトリーがアップロードされていない場合に、ルートを作成	● ユーザーの熟練度に関わらず、手順通りにテレメトリーをアップロードしルートを作成できるよう誘導する UI を開発する

			できないエラーが発生した	
	サービス運用	ビジネスモデルの改善	Linux システム、コマンドラインインターフェースを理解しているエンジニアでないと操作が難しいため、運用が困難	GUI を導入し、エンジニア以外のユーザーでも操作しやすい UI を開発する
		販路拡大	建設、工事現場などでの現場環境の変化に弱く、都度現場情報のアップロードが必要なため運用が困難	オペレーションマニュアルを作成し、自動運転車両の運用周りの環境整備、作業員への周知徹底を実施する
ドローン	システム（機能）	障害物・構造物回避性能向上	ルート作成時、障害物や構造物に影響されないルート作成には微妙な調整が必要	3D 都市モデルの情報（ランドマーク情報、緊急輸送道路など）を用いて、飛行が禁止されている構造物や道路上空などに誤って設定された飛行ルートを、AI 等の活用により自動で OK/NG 判定を行う機能や、自動で経路の修正を行う機能などを追加実装する - 障害物回避アルゴリズムの強化する
		モデルの更新	整備年度等の理由から 3D 都市モデルと実際の周辺環境が異なる場合に、自己位置推定精度が低下し自律運航が困難	3D 都市モデルと実際の周辺環境が異なる場合に、3D 都市モデル由来の事前マップを更新する機能を開発する
		着陸機能	水平・垂直輸送後、帰還する機能が未実装であり、輸送の実現に向けて帰還機能の実装が必要	3D 都市モデルと BIM を組合せたマップを参照しながら自動搬送車両（又はドローン基地）に戻り着陸する機能を開発する

10-3. 今後の展望

本プロジェクトでは、車両・ドローンのルート作成・運航管理を一つのシステムに統合したことで、特定条件下での水平・垂直方向の自動運航が可能になったほか、ユーザビリティの向上を実現することができた。また、車両においては、RTK-GNSS 測位のデータを利用して点群マッチングから得た自己位置を補完することにより、自己位置推定精度が向上し安定した走行を実現できることが分かった。

今後は、社会実装を見据えて、現場で利用できるレベルまで更に UI/UX を向上させる必要がある。具体的には、手動編集機能を備えた復路ルートの自動生成機能の開発や、真上・真横表示への対応などの視覚的な経路設定の改善を目指す。さらに、建設現場での利用を想定し、工事の進行によって日々更新される現場環境をマップデータに容易に反映できる手法、もしくはノイズを除去しながら自己位置推定精度を保つ手法を検討する必要がある。

これらの取組みにより、建設現場における資材運搬のラストワンマイル輸送の自動化を実現し、省人化・効率化に貢献する自律運航システムの社会実装を目指す。

11. 用語集

A) アルファベット順

表 11-1 用語集（アルファベット順）

No.	用語	説明
1	Autoware	・ Linux と ROS2 をベースとした自動運転システム用オープンソースソフトウェアで、レーザレーダ、カメラ、GNSS などの環境センサーを利用し、自車位置や周囲物体を認識しながら、走行ルート上を自律走行する。
2	CloudCompare	・ 3D ポイントクラウド処理ソフトウェア
3	ELAS	・ センシンロボティクス社が提供する Edge LiDAR Autonomous System の略称 ・ 【SL016】の記載と重複するため割愛
4	Galileo	・ EU による全地球航法衛星システム
5	GCS	・ Ground Control System の略 ・ 無人航空機を人間が制御するためのシステム
6	GLONASS	・ 旧ソビエト連邦が開発し、現在はロシア航空宇宙軍の手によってロシア政府のために運用されている衛星測位システム
7	GNSS	・ Global Navigation Satellite System の略 ・ 「全球測位衛星システム」を意味する ・ 米国の GPS、日本の準天頂衛星（QZSS）、ロシアの GLONASS、EU の Galileo 等の衛星測位システムの総称
8	IMU	・ Inertial Measurement Unit の略 ・ 慣性計測装置のことで 3 次元の慣性運動（直行 3 軸方向の並進運動及び回転運動）を検出する
9	LiDAR	・ Light Detection And Ranging の略 ・ レーザー光を照射して、その反射光の情報を基に対象物までの距離や対象物の形などを計測する技術
10	MAVLink	・ ドローンと GCS の間で機体の状態やルート情報などのデータを送受信するプロトコル。メッセージフォーマットを XML で定義することで C++ や Python のヘッダファイルを自動的に生成する
11	MQTT	・ メッセージ指向ミドルウェアのアプリケーション層で使用する、TCP/IP による Pub/Sub 型データ配信モデルの軽量なデータ配信プロトコル
12	NDT	・ 環境地図をボクセルごとに区切り正規分布を計算し 3 次元 LiDAR から得られた点群データとマッチングを行う手法

13	NMEA	・ GNSS 受信機から測位結果として出力されるデータ形式の一つ ・ NMEA0183 というフォーマット（NMEA フォーマット）がある
14	PCD	・ Point Cloud Data の略 ・ 点群を扱うファイル形式の一つ
15	ROS	・ ロボット用のソフトウェアプラットフォーム Robot Operating System の略 ・ ロボット開発において重要な役割を果たすオープンソースのロボット制御ソフトウェア、及びそれを包括するロボット開発プラットフォーム全体のこと
16	RTK	・ Real Time Kinematic の略 ・ 固定局と移動局の 2 つの受信機を利用し、リアルタイムに 2 点間で情報をやり取りすることで、高精度での測位を可能にする手法
17	SENSYN CORE Pilot	・ センシンロボティクス製の GCS アプリ
18	State machine	・ 複雑な命令を与え、制御するための手法 ・ いくつかの「状態（State）」と、ある状態から別の状態への「遷移（Transition）」で構成されており、遷移が起こる条件を満たした際に状態が変わる仕組みになっている
19	Ubuntu	・ デスクトップ PC やクラウド、インターネットに接続されたあらゆる機器まで、全ての環境において動作可能なオープンソースのソフトウェアオペレーティングシステム
20	Unity	・ Unity Technologies が開発・販売している、統合開発環境を内蔵するゲームエンジン

B) 五十音順

表 11-2 用語集（五十音順）

No.	用語	説明
1	ウェイポイント	・経路上の地点情報 ・自動運航を行う際に、現在地から次のポイント（座標）を設定したポイント ・複数のウェイポイントを設定することで、ゴールまでの運航ルートをつなぐことが可能
2	ウェイポイントアクション	・ウェイポイントとして設定したポイント上で、ドローンの方向転換などを行うアクション
3	ガウシアンフィルター	・ガウス分布に基づく重み付け平均化を行うことで、データを平滑化するフィルター
4	基準座標系	・ELAS 独自の 3 次元直交座標系
5	自己位置推定	・ある場所にいる自分自身の位置を推定する技術 ・GPS を使う手法やカメラ映像を使う手法、LiDAR や超音波センサーを使用する手法などが存在する
6	ビジュアルイナーシャルオドメトリ（VIO）	・連続した画像の間に起こる変化、変移を捉えることで算出した移動体の移動量に加え、IMU 情報を加味して移動量を推定する技術
7	ラストワンマイル	・現在は、物流・交通業界において多く用いられ、「顧客にモノ・サービスが到達する最後の 1 マイル（1.6km 程度）の区間・接点」を指す ・元々は通信業界に用いられていた用語で「生活者や企業に対し、通信接続を提供する最後の区間・接点」を意味していた

以上

3D 都市モデルと BIM を活用したモビリティ自律運航システム v3.0
技術検証レポート

発 行：2025 年 3 月

委託者：国土交通省 都市局

受託者：株式会社竹中工務店

株式会社センシンロボティクス

アダワープジャパン株式会社